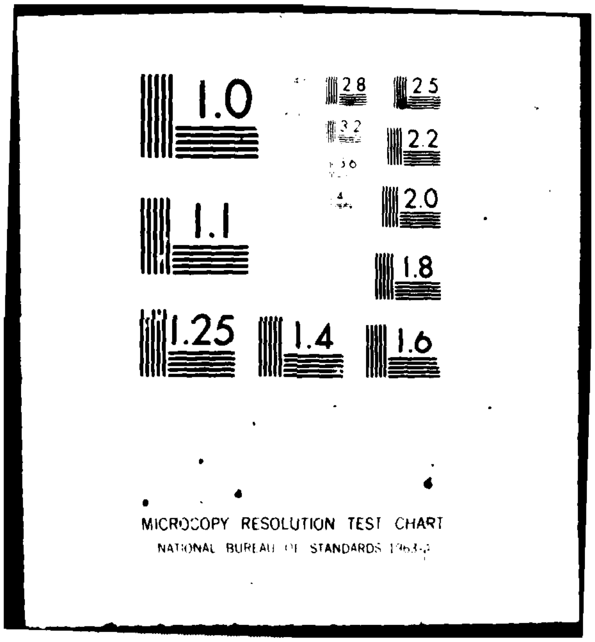
AD-A084 236 VIRGINIA UNIV CHARLOTTESVILLE DEPT OF APPLIED MATHEM--ETC F/G 12/1
ANALYSIS OF COMPUTATIONAL METHODS FOR NONLINEAR PARABOLIC DIFFE--ETC(U)
APR 80 R L BROWN AFOSR-79-0024
UNCLASSIFIED UVA/525629/AMC80/101 AFOSR-TR-80-0340 NL

1 OF 1
AD
A084236

END
DATE
FILMED
6-80
DTIC


1.0
2.8
2.5
3.2
2.2
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

AFOSR-TR- 80 - 0340

LEVEL II

AD A084236

RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES

SCHOOL OF ENGINEERING AND APPLIED SCIENCE

AFOSR-79-0024

UNIVERSITY OF VIRGINIA

Charlottesville, Virginia 22901

A Final Report

ANALYSIS OF COMPUTATIONAL METHODS FOR NONLINEAR PARABOLIC DIFFERENTIAL SYSTEMS

Submitted to:

Air Force Office of Scientific Research
Bolling Air Force Base
Washington, D. C. 20332

DTIC ELECTE MAY 19 1980 S C D

Submitted by:

R. Leonard Brown
Assistant Professor

Report No. UVA/525629/AMCS80/101

April 1980

80 5 14 072

RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES

Members of the faculty who teach at the undergraduate and graduate levels and a number of professional engineers and scientists whose primary activity is research generate and conduct the investigations that make up the school's research program. The School of Engineering and Applied Science of the University of Virginia believes that research goes hand in hand with teaching. Early in the development of its graduate training program, the School recognized that men and women engaged in research should be as free as possible of the administrative duties involved in sponsored research. In 1959, therefore, the Research Laboratories for the Engineering Sciences (RLES) was established and assigned the administrative responsibility for such research within the School.

The director of RLES—himself a faculty member and researcher—maintains familiarity with the support requirements of the research under way. He is aided by an Academic Advisory Committee made up of a faculty representative from each academic department of the School. This Committee serves to inform RLES of the needs and perspectives of the research program.

In addition to administrative support, RLES is charged with providing certain technical assistance. Because it is not practical for each department to become self-sufficient in all phases of the supporting technology essential to present-day research, RLES makes services available through the following support groups: Machine Shop, Instrumentation, Facilities Services, Publications (including photographic facilities), and Computer Terminal Maintenance.

Final Report

ANALYSIS OF COMPUTATIONAL METHODS FOR NONLINEAR PARABOLIC DIFFERENTIAL SYSTEMS.

Submitted to:

Air Force Office of Scientific Research
Bolling Air Force Base
Washington, D. C. 20332

Submitted by:

R. Leonard Brown*
Assistant Professor

11 Apr 80

Department of Applied Mathematics and Computer Science
RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

*Current Title and Address:

Assistant Professor of Computer Science
Mathematical Science Department
Drexel University
Philadelphia, PA 19104

Report No. UVA/525629/AMCS80/101

April 1980

Copy No. 7

Acknowledgments

The research reported herein is based on my exposure to problems in modeling reentry vehicles which occured during a Summer appointment as a USAF-ASEE Fellow. Dr. John C. Adams, Jr., of the VonKarmann Facility at Arnold Engineering Development Center, showed me what needs to be done, how it is being done now, and encouraged me to look for better ways to model reentry vehicles. With financial support from AFOSR, I have made a first pass at finding out why the numerical solution of the nonlinear equations involved in flow on a cone is so difficult, and am now in a position to do more on improving the behavior of the numerical solutions. I could not have done so much without the help of two graduate students, Kurt R. Kovach and Jeffrey L. Popyack. Also, I moved from the University of Virginia to Drexel University in August, 1979, and Drexel has continued to support this research with several thousands of dollars in computer funds.

1. PROBLEM DESCRIPTION

Many important problems in fluid dynamics, among other areas, are modeled by nonlinear parabolic differential systems with initial values given in one "independent variable" x, and boundary values in the remaining dependent variables. Hyperbolic systems can sometimes be treated as a special case. For example, the inviscid flow case of the Navier-Stokes equations [1] is a hyperbolic system, which the viscous flow case is elliptical. A survey of currently used numerical methods is in Richtmeyer and Morton [2]. In subsonic flow cases, the nonlinear terms are small enough to be ignored, but these terms must be included in supersonic and hypersonic flow. These numerical calculations usually involve a finite difference mesh over the boundary value problem variables, resulting in a space discretization matrix equation which for the nonlinear system varies at each step in x, the independent variable representing time in the dynamic case or one of the space variable for the steady state case. Then this nonlinear system is solved as an initial value problem in x. The initial value problem is usually solved by a one step implicit method for reasons of cost and stability. Some methods based on finite element methods for the boundary value problem can be used, but successful methods are only available for the linear cases, such as subsonic flow problems [3].

All of these methods require large amounts of computer memory to store the matrices, and, particularly in the nonlinear case where the matrices must be reevaluated often, large amounts of time. Therefore, it is desirable to investigate the relationship of various aspects of these numerical methods in an effort to reduce the total computation time with no loss in accuracy or significant increase in storage requirements. To give an overview of the current state of development, a sample problem which has been studied by the principal investigator [4] is described.

The incompressible fluid flow around a cone at hypersonic speed and angle of attack $\alpha \geq 0$ is modeled by a parabolic system of nonlinear partial differential equations expressing conservation of energy, mass, and momentum, plus an algebraic equation of state. Typical flow variables are functions of density, velocity and energy. The asymptotic (steady-state) solution in three dimensions is sought. A suitable coordinate system for a cone shaped object uses the variables x, the length from the tip along the cone generator; η the normal to the surface relative to the bow shock stand-off distance ($\eta = \xi/d(x,\phi)$ where ξ is perpendicular to x and d is the bow shock stand-off distance computed from theory); and $\phi = 180°$ at the leeward side. Separation is likely to occur at the leeward side at significant angle of attack $\alpha > 0$, and standard numerical methods have proven inadequate to model this case, so special computer methods have been developed for it. See Figure 1.

Lubard and Helliwell [5] have treated this problem as a parabolic boundary value problem in ϕ and η since theoretical results are available on the behavior of the bow shock, and as an initial value problem in x which allows a marching type numerical solution to be generated given an initial condition away from the point $x = 0$. They treated the nonlinear system

$$\frac{\partial U}{\partial x} + \frac{\partial F(U)}{\partial \eta} + \frac{\partial G(U)}{\partial \phi} = \frac{\partial V(U,\ \partial U/\partial \eta,\ \partial U/\partial \phi)}{\partial \eta} + \frac{\partial W(U,\ \partial U/\partial \eta,\ \partial U/\partial \phi)}{\partial \phi} \qquad (1)$$

where U is the m-dimensional state vector, F and G are given vector

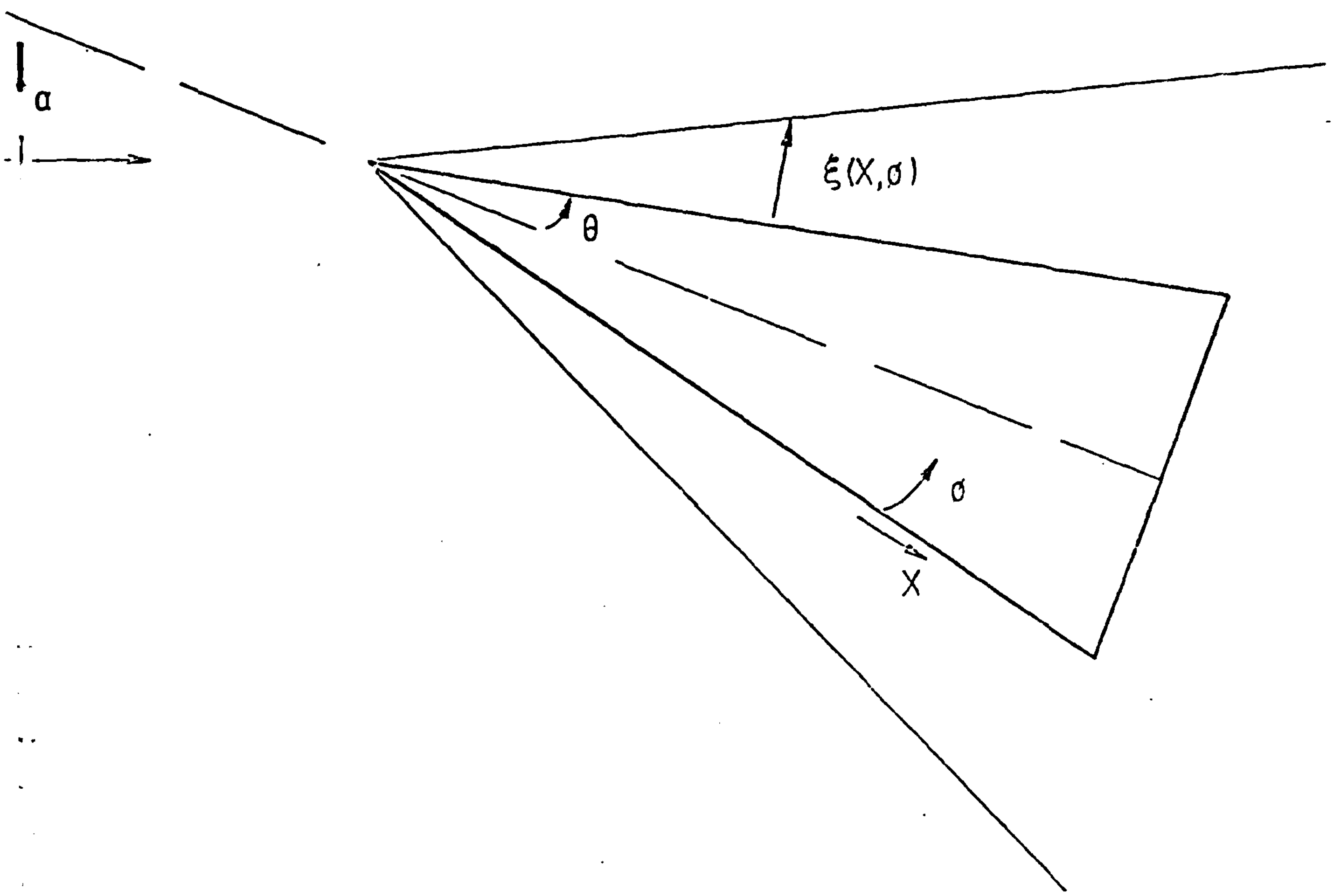


Fig. 1 Coordinate System

functions of U, and V and W are vector functions of U and its partial derivatives. Steady state Navier-Stokes equations are a special case of (1). For the purpose of illustration, let the right hand side of (1) be 0, giving the equation for inviscid flow. A discussion of boundary and initial conditions can be found in [5]. By using a finite difference scheme in η and ϕ for the boundary value problem, a one-step implicit integration scheme in x can be employed to solve the initial value problem. For the implementation of Lubard and Helliwell [5], central differences at each x value

$$\frac{\partial U}{\partial \eta} = \frac{1}{2\Delta\eta}\left(U(\eta_{j+1}, \phi_k) - U(\eta_{j-1}, \phi_k)\right) \tag{2}$$

$$\frac{\partial^2 U}{\partial \phi^2} = \frac{1}{\Delta\phi^2}\left(U(\eta_j, \phi_{k-1}) - 2U(\eta_j, \phi_k) + U(\eta_j, \phi_{k+1})\right) \tag{3}$$

are used with the Backward Euler implicit formula

$$U(x_{i+1}, \eta_j, \phi_k) = U(x_i, \eta_j, \phi_k) + \Delta x\, \partial U(x_{i+1}, \eta_j, \phi_k)/\partial x. \tag{4}$$

Stability considerations derived from considering the numerical solution of an associated linearized system of equations leads to both lower and upper bounds on Δx as a function of $\Delta\eta$ and $\Delta\phi$.

To understand the implementation, consider the linearized problem for $A = \partial F/\partial U$ and $B = \partial G/\partial U$ given as

$$\frac{\partial U}{\partial x} + A\frac{\partial U}{\partial \eta} + B\frac{\partial U}{\partial \phi} = 0. \tag{5}$$

Applying the trapezoidal rule $U(x_{i+1}) = U(x_i) + \frac{\Delta x}{2}\left(\frac{\partial U(x_{i+1})}{\partial x} + \frac{\partial U(x_i)}{\partial x}\right)$, a more accurate implicit scheme than Backward Euler, and using a truncated Taylor Series for F(U) and G(U) given by

$$F(U(x_{i+1})) = F^{i+1} = F^i + A^i \; (U(x_{i+1}) - U(x_i)) + O(\Delta x^2)$$

$$G(U(x_{i+1})) = G^{i+1} = G^i + B^i \; (U(x_{i+1}) - U(x_i)) + O(\Delta x^2)$$

yields the system of linear equations

$$\left[I + \frac{\Delta x}{2} \left(\frac{\partial A^i}{\partial \eta} + \frac{\partial B^i}{\partial \phi} \right) \right] U^{i+1} = \left[I + \frac{\Delta x}{2} \left(\frac{\partial A^i}{\partial \eta} + \frac{\partial B^i}{\partial \phi} \right) \right] U^i - \Delta x \left(\frac{\partial F}{\partial \eta} + \frac{\partial G}{\partial \phi} \right) . \qquad (6)$$

This large, sparse system can be solved by methods such as the alternating direction implicit (ADI) method of Douglas [6] which solves only the equation in η first, then the equations in ϕ. Beam and Warming [7] introduce an error of $(\Delta x)^3$ in an approximate factorization scheme based on Peaceman and Rachford [8] by replacing (6) with

$$\left(I + \frac{\Delta x}{2} \frac{\partial A^i}{\partial \eta} \right) \left(I + \frac{\Delta x}{2} \frac{\partial B^i}{\partial \phi} \right) U^{i+1} = \left(I + \frac{\Delta x}{2} \frac{\partial A^i}{\partial \eta} \right) \left(I + \frac{\Delta x}{2} \frac{\partial B^i}{\partial \phi} \right) U^i$$

$$- \Delta x \left(\frac{\partial F}{\partial \eta} + \frac{\partial G}{\partial \phi} \right) + O(\Delta x^3) . \qquad (7)$$

Since the error introduced is of the same order Δx^3 as the error in the trapezoidal rule, stability is not affected. This equation can then be solved in two levels

$$\left(I + \frac{\Delta x}{2} \frac{\partial A^i}{\partial \eta} \right) \Delta U^* = - \Delta x \left(\frac{\partial F}{\partial \eta} + \frac{\partial G}{\partial \phi} \right) ,$$

$$\left(I + \frac{\Delta x}{2} \frac{\partial B^i}{\partial \phi} \right) \Delta U^i = \Delta U^*$$

$$U^{i+1} = U^i + \Delta U^i .$$

This method has several disadvantages. If used with a more accurate difference method, the error introduced in the factorization will lower the error order of the method; however, if used with a lower order method such as the Backward Euler, good results could be expected.

However, Lubard and Helliwell note that the difference in ϕ near $\eta = 0$ has a singularity there, and use instead a method that uses the factorization (7) in η but solves for each set of solutions at each ϕ_k in sequence $\phi_0 = 0°$ to $\phi_k = 180°$ in steps of $\Delta\phi$. This is done iteratively until the computation converges. At each ϕ_k, the resulting system of linear equations is an n by n block tridiagonal matrix of block size m by m where $i = 1, \ldots, n$ for η_i and $m = 6$, the number of states at each point. The method is comparable to a Gauss-Seidel iteration with each element of the solution replaced by a $(n*m)^2$ size linear equation.

In actual practice, the equations are rewritten to compute the change ΔU in the current value of $U(x_{j+1})$. This is called the delta form of the corrector and yields a linear block tridiagonal system

$$\begin{bmatrix} B_1 & C_1 & & \\ A_2 & B_2 & C_2 & \\ & \ddots & A_n & B_n \end{bmatrix} \Delta U = RHS \tag{8}$$

where A_i, B_i, and C_i are square m by m matrices. RHS is a m by n corrector for $U(x_{i+1})$.

2. Research Conducted

A portable program called HVSL [9], which is a modified version of the Lubard and Helliwell code and is used at Arnold Engineering Development Center (AEDC), was available for experimentation. It is portable since the initial state of the system is read in, in part, from cards and then, after solving a boundary value problem, all initial values at 50 η points and 19 ϕ points are known. These can be changed by an interpolating procedure included in the code. For experimental purposes, the number of ϕ values from windward (0°) to leeward (180°) was reduced from 19 to 3. Validation tests were run at angle of attack $\alpha = 1°$ to determine if this modified system produced the same solution. At least three decimal place agreement was observed at $\phi = 0°$, 90°, and 180°, so it was concluded that this much less expensive test program was adequate for testing modifications to HVSL.

The following changes were made to HVSL.

1. The initial (predicted) value of $U(x_{j+1})$, representing the solution $U(x_{j+1}, \eta_k, \phi_\ell)$ for $k = 1$, NK, $\ell = 1$, NL by the Euler explicit method, is computed by

$$U(x_{j+1}) = U(x_j) + (U(x_j) - U(x_{j-1})) * \Delta x_{j+1}/\Delta x_j .$$

While this appears to be using a finite difference to approximate the derivative $\dot{U}(x_j) = dU(x_j)/dx$, it is actually the correct value since, if Euler's implicit formula is iterated to convergence,

$$U(x_j) = U(x_{j-1}) + \Delta x_j \, \dot{U}(x_j). \tag{9}$$

The modified program stores the derivative term $\Delta x_j \dot{U}(x_j)$ in ΔU^0 after the last evaluation and correction of whatever numerical method is in use, and this is used in an Euler explicit predictor. No additional storage is required.

2. The check for convergence in subroutine IMPETA checks to see if the right hand side of the matrix equation (8) satisfies

$$\sum_{i=1}^{6} RHS^2_{i,j,k,l} \leq 6*10^{-6}$$

for all η_k, for each ϕ_ℓ, $\ell = 1$, NL, for convergence at x_j. The subscript i refers to the six state variables. In test runs, no calculation ever terminated due to meeting this convergence test, but instead the maximum number of iterations were used. A more appropriate convergence criterion would be to stop when the last corrector did not make any changes in the third decimal place of any variable, and this relative change criterion was implemented.

3. The Lubard and Helliwell code uses the Backward Euler corrector

$$U(x_{j+1}) = U(x_j) + \Delta x \, \dot{U}(x_{j+1})$$

to calculate successively better approximations to $U(x_{j+1})$. Using the the delta form, it is seen that

$$\Delta U^0 = U^0(x_{j+1}) - U(x_j) = \Delta x \, \dot{U}(x_j) \tag{10}$$

$$\begin{aligned} \Delta U^1 &= U^1(x_{j+1}) - U^0(x_{j+1}) \\ &= U(x_j) + \Delta x \, \dot{U}^0(x_{j+1}) - (U(x_j) + \Delta x \, \dot{U}(x_j)) \\ &= \Delta x(\dot{U}^0(x_{j+1}) - \dot{U}(x_j)) \end{aligned} \tag{11}$$

$$\begin{aligned} \Delta U^i &= U^i(x_{j+1}) - U^{i-1}(x_{j+1}) \\ &= \Delta x(\dot{U}^{i-1}(x_{j+1}) - \dot{U}^{i-2}(x_{j+1})), \quad i = 2, \ldots, 5. \end{aligned} \tag{12}$$

Since this calculation is already programmed, ΔU can be used as is in two different corrector formulas. The Trapzoidal Corrector

$$U^i(x_{j+1}) = U(x_j) + \Delta x(\dot{U}^{i-1}(x_j) + \dot{U}(x_j))/2.$$

can be implemented by

$$\begin{aligned} U^1(x_{j+1}) &= U(x_j) + \Delta x(\dot{U}^0(x_{j+1}) + \dot{U}(x_j))/2. \\ &= U(x_j) + \Delta x\, \dot{U}(x_j) + \Delta x(\dot{U}^0(x_{j+1}) - \dot{U}(x_j))/2. \\ &= U^0(x_{j+1}) + \Delta U^1/2. \qquad (13) \end{aligned}$$

and then

$$\begin{aligned} U^i(x_{j+1}) &= U(x_j) + \Delta x(\dot{U}^{i-1}(x_{j+1}) + \dot{U}(x_j))/2. \\ &= U(x_j) + \Delta x(\dot{U}^{i-2}(x_{j+1}) + \dot{U}(x_j))/2 + \Delta x(\dot{U}^{i-1}(x_{j+1}) - \dot{U}^{i-2}(x_{j+1})/2. \\ &= U^{i-1}(x_{j+1}) + \Delta U^i/2. \qquad (14) \end{aligned}$$

The Iterated Multistep Method (IMS) due to Hyman [10] is:

$$\begin{aligned} U^0(x_{j+1}) &= U(x_j) + \Delta x\, \dot{U}(x_j) \\ U^1(x_{j+1}) &= U(x_j) + \Delta x(\dot{U}^0(x_{j+1}) + \dot{U}(x_j))/2. \\ U^i(x_{j+1}) &= U^{i-1}(x_{j+1}) + \Delta x(\dot{U}^{i-1}(x_{j+1}) - \dot{U}^{i-2}(x_{j+1}))/(i+1), \quad i = 2,3,4,\ldots. \end{aligned}$$

This can be formulated the same as the Trapezoidal method for $i = 0,1$, and then

$$U^i(x_{j+1}) = U^{i-1}(x_{j+1}) + (\Delta U^i)/(i+1), \quad i = 2,3,\ldots. \qquad (15)$$

These alternative methods have both stability and accuracy advantages over the implicit Euler corrector. The Trapezoidal corrector, when applied to the <u>linear</u> complex equation

$$\dot{U} = \lambda U, \tag{16}$$

with nonzero initial value of U_0, damps out any error introduced by either machine roundoff error or the discretization of the solution with respect to x for any $\Delta x > 0$ as long as λ has a negative real part. This is called A-stability [11]. The exact solution to (16), $\exp(\lambda x)U_0$, behaves the same way since an initial error d_0 yields the solution $\exp(\lambda x)U_0 + \exp(\lambda x)d_0$, and thus the error contribution goes to zero as x goes to infinity if λ has negative real part. This assumes that the Trapezoidal corrector is solved exactly, which is possible in the linear case since

$$U_{i+1} = (1 + \Delta x\lambda/2)/(1 - \Delta x\lambda/2)\ U_i.$$

Stability behavior is somewhat different if U_{i+1} is solved iteratively, as must be the case in a nonlinear equation. Thus, the stability behavior of the trapezoidal corrector should be investigated further.

The accuracy of the Trapezoidal corrector is based on the local discretization error, which is the size of the error in U_{i+1} if U_i were the correct solution. This is proportional to Δx^2 for the implicit Euler corrector, but to Δx^3 for the Trapezoidal corrector. Thus, the Trapezoidal formula is more accurate.

The IMS method, applied to (16), has the property that each successive corrector iteration increases the stability region, i.e. $\Delta x\lambda$ such that errors introduced are not increasing in size as the calculation proceeds.

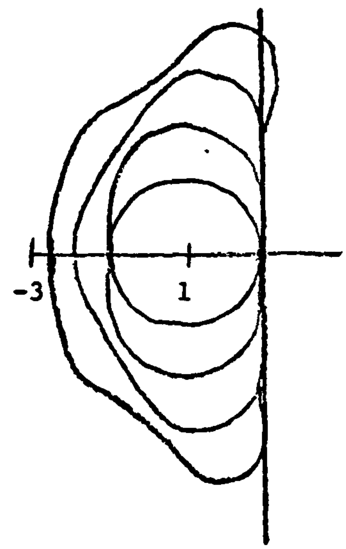


Figure 2. Linear Stability analysis of Iterated multistep method. Consecutively larger figures are U^i for $i = 0,1,2,3$.

See Figure 2. Also, in the linear case only, each application of the IMS corrector equation increases the accuracy, i.e. the discretization error of U^i is proportional to Δx^{i+1}. In the nonlinear case, the error term is similar to that of the Trapezoidal corrector.

The above changes were made to the HVSL test program, and the resulting values were compared to the original program. It was noted that none of the test cases achieved convergence, either the old or new convergence criterion. However, all significant numerical values did agree to two decimal places, so it was concluded that there was a marginal stability problem, and the stability analysis in [12] was insufficient to explain the phenomenon since that analysis was based on linearizing the system and inspecting the eigenvalues of the resulting Jacobian matrices. Therefore, a simpler test case involving only one space variable and time as the independent variable was used to study the three methods. The nonlinear problem has properties similar to the HVSL problem, and uses the same discretization as the Lubard and Helliwell method.

The quasi-one-dimensional time dependent flow of an inviscid perfect gas through a converging-diverging nozzle use the variables:

x = distance, normalized to [0.,1.],

$A(x)$ = nozzle cross-sectional area,

$$U = \begin{bmatrix} \rho \\ \bar{m} \\ \bar{e} \end{bmatrix}$$

where the state variables are ρ, the gas density, $\bar{m} = \rho U$ where U is the velocity along the x axis, and $\bar{e} = \rho(e + U^2/2)$ for $e = c_v T$, where T is the temperature and c_v is the gas constant. Thus, $T = (e - m^2/(2\rho))/(\rho c_v)$.

The equations are

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = G \tag{17}$$

where

$$F = \begin{bmatrix} \bar{m} \\ \rho RT + \bar{m}^2/\rho \\ \bar{m}\ \bar{e}/\rho + \bar{m}RT \end{bmatrix}$$

and

$$G = \begin{bmatrix} -\ \bar{m}\ d(\ln A)/dx \\ -\ \bar{m}^2\ d(\ln A)/dx/\rho \\ -\ \bar{m}(\bar{e}/\rho + RT)\ d(\ln A)/dx \end{bmatrix}$$

A working version of this program was provided by Dr. John C. Adams of AEDC. The program linearized (17) with respect to t, giving

$$\frac{dU}{dt} + \frac{\partial F}{\partial U}\frac{dU}{dx} = G \tag{18}$$

and replacing $\frac{\partial U}{\partial x}$ by finite difference approximations over a net of 101 equally spaced points. Thus, the result is a system of 303 ordinary differential equations in t, with algebraic boundary condition consistent with the method of characteristics solution in one dimension at the inlet, and extrapolation of supersonic outflow at the exit. The resulting block tridiagonal system is

$$\begin{bmatrix} B_1 & C_1 & & \\ A_2 & B_2 & C_2 & \\ & & \ddots & \\ & & A_n & B_n \end{bmatrix} \Delta U = RHS$$

for 3 by 3 matrices A_i, B_i, C_i, $i = 1, \ldots, 101$. The linearized initial value problem is solved exactly, once each time step. The numerical method is parametrized as [13]

$$\frac{dU(t_n)}{dt} = \frac{1}{\Delta t} \frac{(1 + z)\ \Delta - z\nabla}{1 + \Theta\Delta} U(t_n)$$

where Δ is the forward difference operator, ∇ is the backward difference operator, and $\Theta = 1$, $z = 0$ yields the exact implicit Euler solution of the linearized problem; $\Theta = 1/2$, $z = 0$ yields the exact trapezoidal solution; and $\Theta = 0$, $z = 0$ would yield the explicit Euler predictor except the program would divide by zero if $\Theta = 0$. This does not emulate the iterative solution technique in the Lubard and Helliwell code, so the program was rewritten to use an explicit Euler predictor approximated by $\Theta = 10^{-r}$, r large, then successive linearizations and computations of ΔU^i consistent with the implicit Euler technique used in HVSL. Then either ΔU^i could be used as is to get the implicit Euler predictor, or else equations (10, 13, 14) for Trapezoidal corrector or (10, 13, 15) for IMS corrector could be used.

The results for the Trapezoidal and IMS test runs, using a constant 4 corrector iterations, agreed to 4 decimal places with the original program at the 4th time step, and to 2 decimal places after 15 time steps (the equations are being integrated to steady state). The velocity U, which depends on $\bar{e}$ and ρ, becomes unstable by the 4th time step when the implicit Euler corrector was iterated 4 times, by the 10th time step when iterated only once. Thus, the stability properties of the linearized equation are seen to be different from those of the nonlinear equation. Note that the step size Δt was chosen to meet the Courant-Friedrich-Lewy criteria for the linearized implicit Euler formula, yet this step size does not work with the nonlinear equation it is based on. This confirms that a stability problem exists.

A new technique has been developed to study stability of ordinary differential equation integrators as they are applied to nonlinear differential systems [14]. The analysis can usually be carried out using an interactive graphics package called STAN. If the user knows an approximate equilibrium point U^* where $dU/dt = 0$, then it is possible to investigate the stability of any two dimensional subsystem by varying only two values, i.e., we investigate the system $u = (u_i, u_j) = U^* + e_i d_i + e_j d_j$ where e_i, e_j are the i-th and j-th unit vectors and d_i, d_j are scalar perturbations. This allows the contractivity region defined by Dahlquist [15] to be mapped. The boundary of this region consists of points at which the forward difference in the independent variable t of a quadratic function $V(u) = u^*Qu$ is zero. V(u) is chosen in the same way as the Liapunov stability function is chosen [16], using the Jacobian matrix of the derivative with respect to the vector $u = (u_i, u_j)$. Thus, if this boundary is found using the exact solution, then any solution $U(t_0 + \Delta t)$ with initial conditions $U(t_0)$ on the boundary has the property $V(u(t_0 + \Delta t)) = V(u(t_0))$. This can be further refined to compute a stability region inside which $u(t_0 + n\Delta t)$ will stay for any n, at least in the autonomous case. Similar regions can be generated for the numerical solutions using the same Δt. Figure (3) illustrates the results for the linear case:

$$y = u_1 + iu_2$$

$$\dot{y} = \lambda y, \quad y(0) = (u_1, u_2) \qquad (19)$$

$$\lambda = \ln(y_0)/\Delta t$$

with exact solution at mesh points $y(t_n) = y_0^{n+1}$. For $V(u) = u_1^2 + u_2^2$, the contractivity region and stability region are both the unit circle about 0.

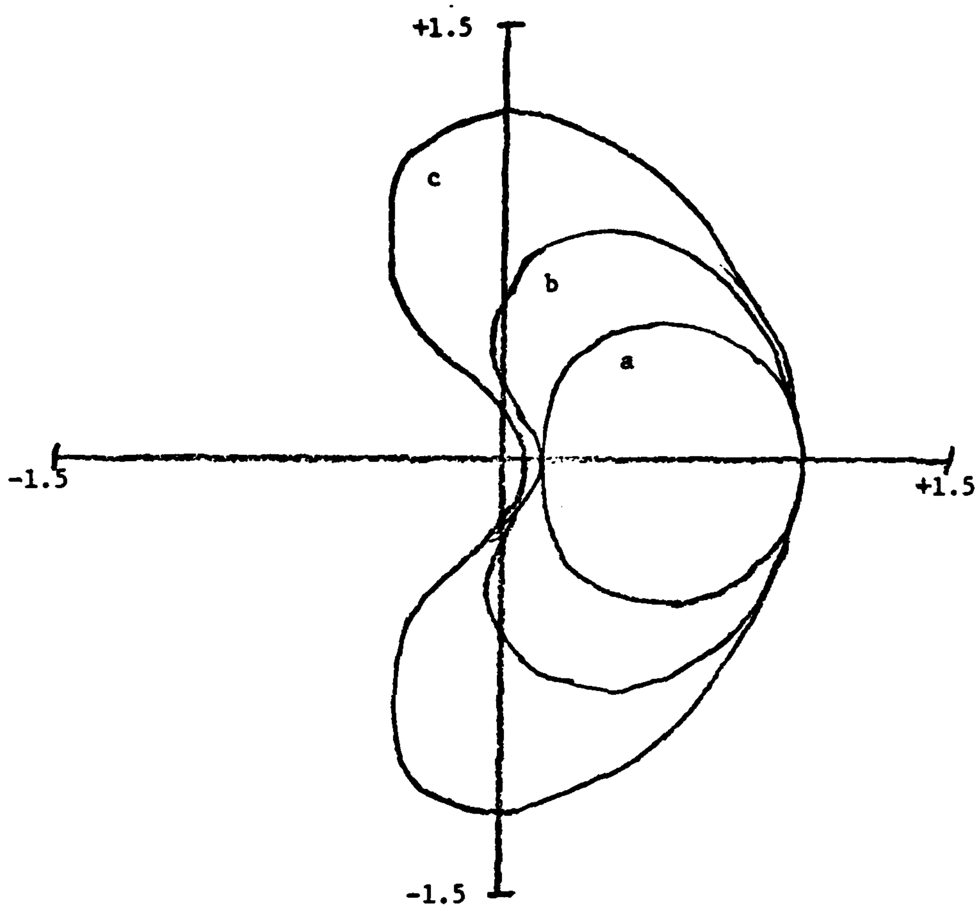


Figure 3A. Non-Linear Stability analysis of y' = λy:
Contractivity region for a) explicit Euler,
b) trapezoidal with one corrector iteration,
c) trapezoidal with 2 corrector iterations.

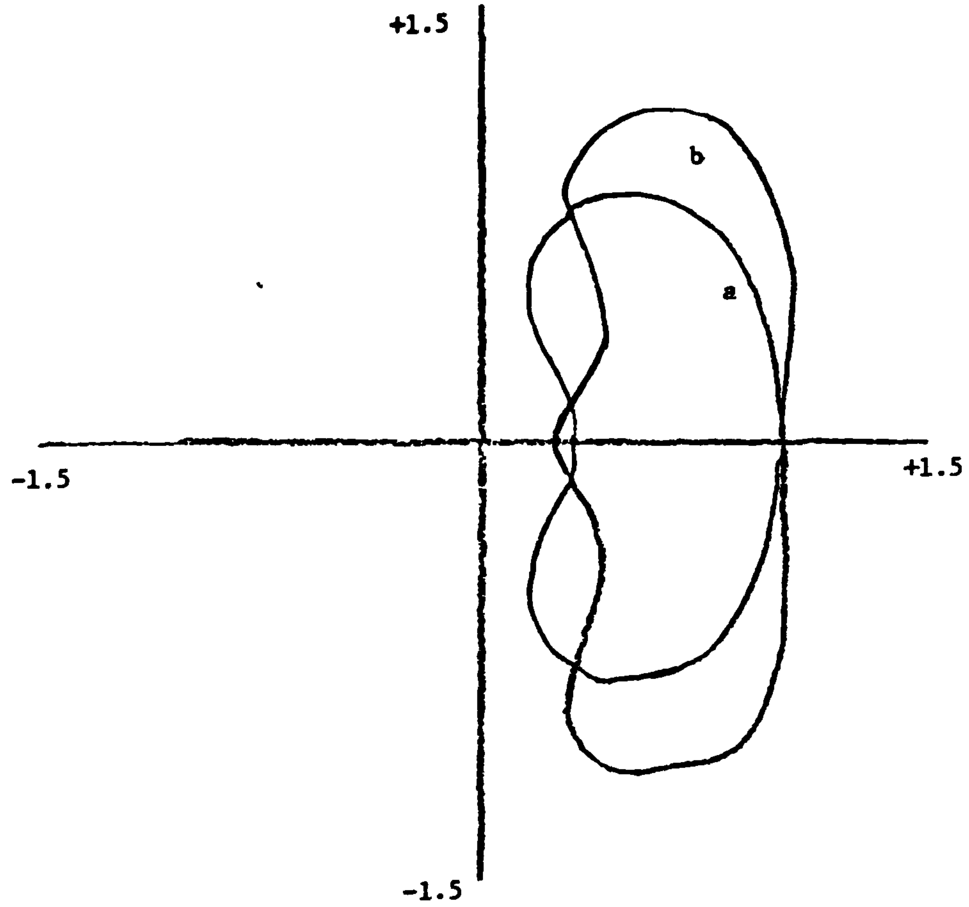


Figure 3B. Non-Linear Stability Analysis of $y' = \lambda y$ a) explicit Euler predictor, one implicit Euler corrector b) same predictor, 2 corrector iterations.

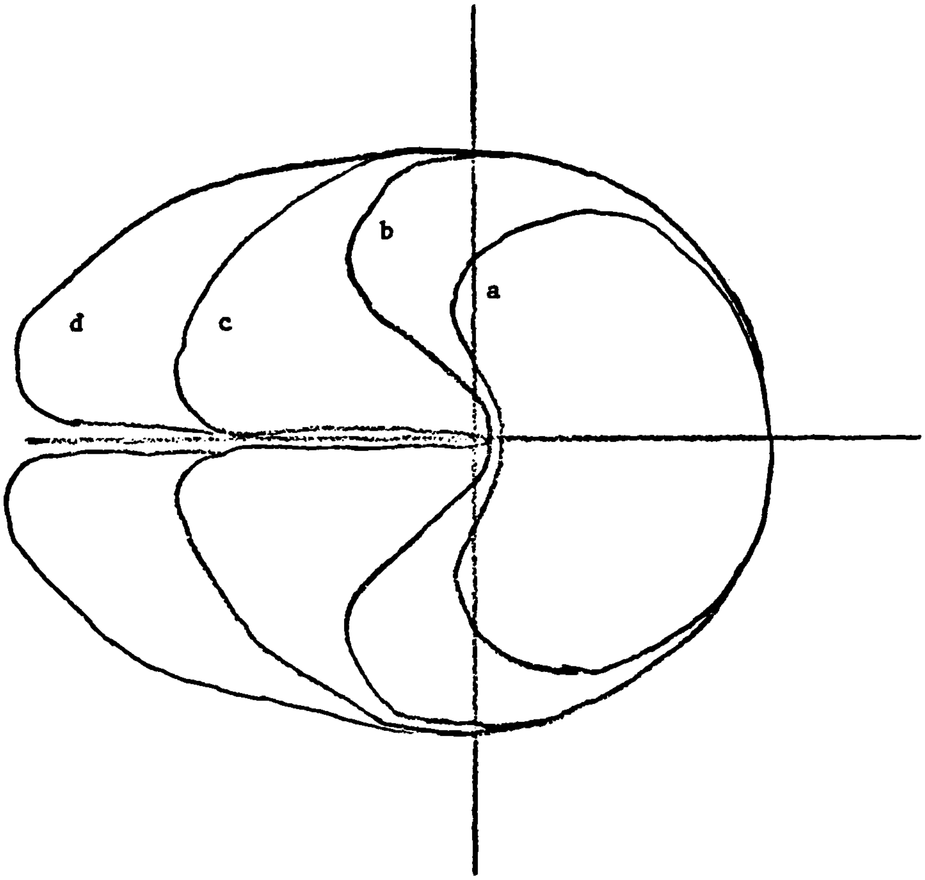


Figure 3C. Non-Linear Stability analysis of $y' = \lambda y$ for the iterated multistep method for a) 2 steps b) 3 steps c) 4 steps d) 5 steps.

The point (0,0) corresponds to the point at infinity in the linear analysis described earlier, and the point (1,0) corresponds to the origin in the linear analysis.

To get the exact solution to a differential system, a program SERIES [17], written in PASCAL, is used. It can accept up to 20 first order ordinary differential equations as input and will write a FORTRAN subroutine SOL(T, Y, YO, IND) which, when called for a certain value of $t = T$, given initial conditions of $u(t)$ at $t = 0$ stored in the FORTRAN array YO, will recursively generate the coefficients of the power series solution of $u(t)$ starting with the constant terms stored in YO, and output in the array Y the solution $u(t)$, or else indicate that the radius of convergence of the series is not greater than t by setting IND to certain nonzero values. Of course, not every possible function is included, but the trigonometric, logorithmic, and exponential functions of the dependent variables are allowed, as well as most FORTRAN functions of the independent variable. This program has been tested on numerous nonlinear systems and the resulting subroutine SOL has been interfaced to STAN, but no system of partial differential equations has been included.

Two different stability analyses were attempted for the converging-diverging nozzle example. In one, two interior stations were isolated, and the forward divided difference was used on all variables that were differentiated with respect to x. Letting two consecutive ρ values be called R1 and R2, two consecutive $\bar{m}$ values be XM1 and XM2, and two consecutive $\bar{e}$ values be E1 and E2, a system of six state variables would result. However, temperature, which could be considered constant but is actually a slowly varying function of $\bar{e}$, $\bar{m}$, and ρ, and $d(\ln A)/dx$, a constant, must be accounted for at the two points. By setting their derivatives with respect to t to zero, these constants can be input to STAN along with the state variables. Let XK1 and XK2 be consecutive values of d(ln A)/dx and TMP1 and TMP2 be temperature values, and denote the derivative of a variable Z by Z•, then the resulting input to SERIES is:

```
R1.=-(XM2-XM1)/DX-XM1*XK1;
R2.=-(XM2-XM1)/DX-XM2*XK2;
XM1.=-XM1*XM1/R1*XK1+((XM1/R1)**2-R*TMP1)*(R2-R1)/DX
-2.*XM1/R1*(XM2-XM1)/DX;
XM2.=-XM2*XM2/R2*XK2+((XM2/R2)**2-R*TMP2)*(R2-R1)/DX
-2.*(XM2*XM2/R2)*(XM2-XM1)/DX;
E1.=-XM1*(E1/R1+R*TMP1)*XK1+(XM1*E1)/R1**2*(R2-R1)/DX
-(E1/R1+R*TMP1)*(XM2-XM1)/DX-XM1/R1*(E2-E1)/DX;
E2.=-XM2*(E2/R2+R*TMP2)*XK2+XM2*E2/R2**2*(R2-R1)/DX
-(E2/R2+R*TMP2)*(XM2-XM1)/DX-XM2/R2*(E2-E1)/DX;
TMP1.=0.;
TMP2.=0.;
XK1.=0.;
XK2.=0.;;
```

where R and DX are constants and the known equilibrium values from a test run can be read in. Since TMP = $(\bar{e} - m^2/2\rho)/\rho c_v$, TMP1. and TMP2. can also be entered by differentiating this expression, but results will be similar. Appendix A contains the output of SERIES for this input.

In order to avoid using the same derivative with respect to x, a system based on one x point with constant input partial derivatives was also tried. This system uses the variables: RX for ρ; XM for $\bar{m}$; E for $\bar{e}$; RDX for $\partial\rho/\partial x$; XMDX for $\partial\bar{m}/\partial x$; and EDX for $\partial\bar{e}/\partial x$; TMP for temperature, and XK for d(ln A)/dx.

```
RX.=-XM*XK-XMDX;
XM.=-XM*XM/RX*XK-(R*TMP-(XM/RX)**2)*RXDX
-2.*XM/RX*XMDX;
E.=-XM*(E/RX+R*TMP)*XK+XM*E/RX**2*RXDX
-(E/RX+R*TMP)*XMDX-XM/R*EDX;
RXDX.=0.;
XMDX.=0.;
EDX.=0.;
TMP.=0.;
XK.=0.;;
```

Appendix B is the output of SERIES for this input.

Both systems were tested against the explicit Euler solution of the corresponding initial value problem (SERIES also generates a FORTRAN subroutine DIFFUN(T, Y, DY) which fills the array DY with the derivative evaluated at u(t) where t = T, u is in array Y). It was discovered that the radius of convergence of the power series contracted sharply for values past the throat of the nozzle, so only values between the inlet and the throat can be analyzed using STAN. Table I gives initial values that were picked for analysis. Note that the throat is at x = .26 where $d(\ln A)/dx = 0$.

Both of these systems were run with both sets of initial data, and the resulting contractivity regions are displayed in Figure 4. These were only achieved for Δt of $5*10^{-12}$, and do not correspond to the expectations of results from test runs. Also, they are identical for both the analytic and numerical solution, which suggests they are actually an artifact of the program STAN. This can be seen to be the case since the first step of generating the stability region about an equilibrium point U* is to

Table I - Values input to STAN. (Values in parenthesis used in two point scheme, --DX used only in one point scheme).

x	R1(R2)	XM1(XM2)	E1(E2)	RDX	XMDX	EDX	TMP1(TMP2)	XK1(XK2)
.06	.063	16.47	355587.	-.0364	28.2	-316100.	1309.	-8.77746
.07	(.0626)	(16.75)	(352426.)				(1305.)	(-9.11899)
.25	.04	56.8	231022.	-.43	10.413	-218570.	1098.	-3.53
.26	(.036)	(57.8)	(209565.)				(1050.)	(0.)

R = 1716.

DX = .01

DT = 5.E-6

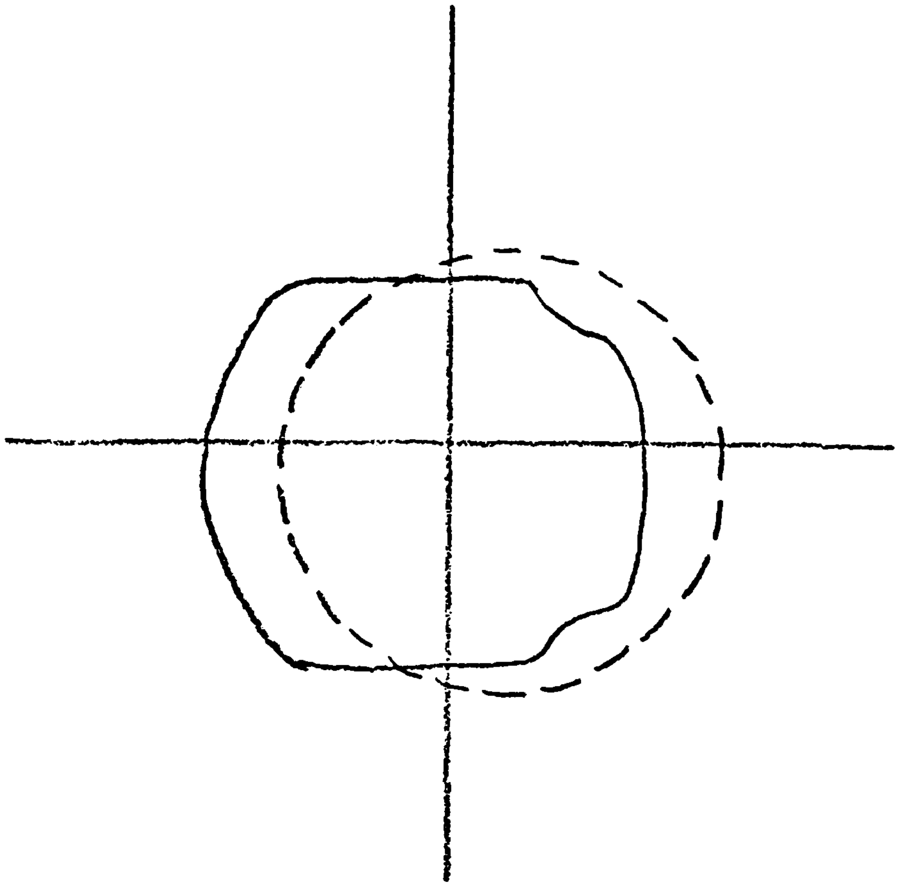


Figure 4. Non-linear stability analysis of the nozzle problem. Solid line: $\bar{m}$ vs. $\bar{e}$ with center at (16.5, 355587.). Dashed line: ρ vs. $\bar{m}$ with center at (0., 16.5). Both axes of length 1.

3. Research Findings

Relative to the specific research outlined in the original proposal on this research, the following findings are of interest.

1. A study of recent literature on finite elements [18],[19] reveals that use of finite element techniques are not easily adaptable to nonlinear systems in several dimensions, especially when the system is designed to be easily changed as different reentry vehicle configurations are tested. The analysis of minimum and maximum step size found in [12] is really inadequate in the finite difference case, depending as it does on linearization of the differencial system, and a similar analysis in the finite element case seemed both beyond the scope of the intended research and not very fruitful.

2. Since the algebraic amplification matrices involved in the linear stability analysis of the methods under investigation did not point out the experimentally observed instability, it was inappropriate to develop a complicated algebraic manipulation package to compute such matrices.

3. After working with the two model packages HVSL and the converging-diverging nozzle, it was concluded that the form in which the updates to the dependent variable vector is derived (the "delta" form), resulting in parts of the numerical method being computed at various stages and places in the program, would make it difficult to submit these codes to use by standard packages, [20], [21] which usually require a single subroutine such as DIFFUN(T, Y, DY) to compute the first derivative array DY given the state variable array Y and the independent variable, T. On the other hand, the "delta form" can be easily adapted to most implicit multi-step correctors

$$U(x_j) = \sum_{i=1}^{k} \alpha_i U(x_{j-i}) + \Delta x \sum_{i=0}^{k} \beta_i \dot{U}(x_{j-i}),$$

search from U^* along a particular line in (u_1, u_2) space emanating from U^* searching for the initial conditions of the bisection method, i.e., find i such that $\Delta V(U^* + 2^i(e_1, e_2))$ has different sign from $\Delta V(U^* + 2^{i+1}(e_1, e_2))$, where (e_1, e_2) represents the unit vector in the search direction for the two variables being changed, u_1 and u_2. The variable i varies from -3 to 4. In most applications of ordinary differential equation stability regions, the various state equations are not highly coupled, and when two equations are coupled, one usually attempts to find the stability region using these two variables. For discretized partial differential equations, however, the variables are necessarily highly coupled, and apparently a change on the order of 100 percent creates an immediate overflow. Thus, the actual stability region for $\Delta t = 5*10^{-12}$ occurs because $\Delta V = 0$ when Δt is made so small that $V(u_1(0),u_2(0)) = V(u_1(\Delta t),u_2(\Delta t))$ by either numerical or analytical techniques.

To test this hypothesis, one value, at x = .24, of ρ, of $\bar{m}$, and of $\bar{e}$ were each changed to 10 times their original value and the original, linearized, converging-diverging nozzle program was run. In each of the three cases, overflow stopped the calculation by the third time step. Therefore, STAN must be modified to take into account the coupling of the state variables when analyzing the stability of systems of partial differential equations. This modification is currently being made. If successful, it would open the way to automatic stability analysis of complicated systems of nonlinear equations to allow researchers to choose which numerical method is most appropriate, from a stability standpoint, to integrate their parabolic-hyperbolic discretized system.

with only the additional storage for carrying the $U(x_{j-i})$, $U^1(x_{j-i})$ terms. Also, methods for changing step size Δx and even changing from one formula to another could be adapted from the packages. However, since the finite differences are usually only first order accurate the effort on nonlinear systems of a high order method in the independent variable versus a low order method in the spatial discretization is not yet understood. Certainly in the linear case there is a stability argument against it.

4. The best tool now available for the analysis of a nonlinear parabolic system could be STAN, provided an understanding of how to reduce a discretized system to one of a convenient size for such an analysis. Certainly the cost of both computer time and programmer time of entering the entire discretized system is prohibitive, yet the two attempts to enter significant subsystems did not yield sufficient information to show the value of this analysis technique.

However, experimental evidence exists that the numerical methods currently used to model high speed flow on a cone is, at best, marginally stable, and the results suspect. Continued research should be undertaken to provide an understandable method for directly analyzing the stability properties of parabolic-hyperbolic systems, and comparing them to the stability behavior of numerically generated solutions, and to chose, when appropriate, more accurate numerical methods that do not require significantly larger storage.

APPENDIX A

Output of SERIES for divided difference formulation

```
      SUBROUTINE DIFFUN(T,Y,DY)
      DIMENSION DY(20),Y(20)
      DATA R/1716./,XK1/7.86711/,XK2/7.59839/,CV/4290./,DX/.01/
      DY(1)=-(Y(4)-Y(3))/DX-Y(3)*XK1
      DY(2)=-(Y(4)-Y(3))/DX-Y(4)*XK2
      DY(3)=-Y(3)*Y(3)/Y(1)*XK1+((Y(3)/Y(1))**2-R*Y(7))*(Y(2)-Y(1))/DX-
     +2.*Y(3)/Y(1)*(Y(4)-Y(3))/DX
      DY(4)=-Y(4)*Y(4)/Y(2)*XK2+((Y(4)/Y(2))**2-R*Y(8))*(Y(2)-Y(1))/DX-
     +2.*(Y(4)/Y(2))*(Y(4)-Y(3))/DX
      DY(5)=-Y(3)*(Y(5)/Y(1)+R*Y(7))*XK1+(Y(3)*Y(5))/Y(1)**2*(Y(2)-Y(1)
     +)/DX-(Y(5)/Y(1)+R*Y(7))*(Y(4)-Y(3))/DX-Y(3)/Y(1)*(Y(6)-Y(5))/DX
      DY(6)=-Y(4)*(Y(6)/Y(2)+R*Y(8))*XK2+Y(4)*Y(6)/Y(2)**2*(Y(2)-Y(1))/
     +DX-(Y(6)/Y(2)+R*Y(8))*(Y(4)-Y(3))/DX-Y(4)/Y(2)*(Y(6)-Y(5))/DX
      DY(7)=(((-Y(3)*(Y(5)/Y(1)+R*Y(7))*XK1+(Y(3)*Y(5))/Y(1)**2*(Y(2)-Y
     +(1))/DX-(Y(5)/Y(1)+R*Y(7))*(Y(4)-Y(3))/DX-Y(3)/Y(1)*(Y(6)-Y(5))/DX
     +)-Y(3)*(-Y(3)*Y(3)/Y(1)*XK1+((Y(3)/Y(1))**2-R*Y(7))*(Y(2)-Y(1))/DX
     +-2.*Y(3)/Y(1)*(Y(4)-Y(3))/DX)/Y(1))/Y(1)-(Y(5)-Y(3)**2/Y(1))*(-(Y(
     +4)-Y(3))/DX-Y(3)*XK1)/Y(1)**2)/CV
      DY(8)=(((-Y(4)*(Y(6)/Y(2)+R*Y(8))*XK2+Y(4)*Y(6)/Y(2)**2*(Y(2)-Y(1
     +))/DX-(Y(6)/Y(2)+R*Y(8))*(Y(4)-Y(3))/DX-Y(4)/Y(2)*(Y(6)-Y(5))/DX)-
     +Y(4)*(-Y(4)*Y(4)/Y(2)*XK2+((Y(4)/Y(2))**2-R*Y(8))*(Y(2)-Y(1))/DX-2
     +.*(Y(4)/Y(2))*(Y(4)-Y(3))/DX)/Y(2))/Y(2)-(Y(6)-Y(4)**2/Y(2))*(-(Y(
     +4)-Y(3))/DX-Y(4)*XK2)/Y(2)**2)/CV
      DY(9)=1.
      RETURN
      END
```

```
      SUBROUTINE SOL(T,YO,YNEW,IND)
      DIMENSION YO(20),YNEW(20),ZZZB(20),DTFAKE(20),DR1(20),R1(20),XM2
     +(20),XM1(20),TST(20),TSU(20),TSV(20),TSW(20),TSX(20),DR2(20),R2
     +(20),TS1(20),TS2(20),DXM1(20),TS3(20),TS4(20),TS5(20),TS6(20),TS7
     +(20),TS8(20),TMP1(20),TS9(20),TTS(20),TTT(20),TTU(20),TTV(20),TTW
     +(20),TTX(20),TTY(20),TT0(20),TT1(20),TT2(20),DXM2(20),TT3(20),TT4
     +(20),TT5(20),TT6(20),TT7(20),TT8(20),TMP2(20),TT9(20),TUS(20),TUU
     +(20),TUV(20),TUW(20),TUY(20),TU0(20),TU1(20),TU2(20),DE1(20),E1
     +(20),TU3(20),TU5(20),TU6(20),TU7(20),TU8(20),TU9(20),TVS(20),TVT
     +(20),TVV(20),TVW(20),TVX(20),TV2(20),TV3(20),TV4(20),E2(20),TV6
     +(20),TV7(20),TV8(20),TV9(20),DE2(20),TWS(20),TWU(20),TWV(20),TWW
     +(20),TWX(20),TWY(20),TWZ(20),TW0(20),TW2(20),TW3(20),TW4(20),TW9
     +(20),TXS(20),TXT(20),TXW(20),TXX(20),TXY(20),TZ6(20),TZ7(20),TZ8
     +(20),TZ9(20),T0S(20),T0T(20),T0U(20),T00(20),T02(20),T03(20),T04
     +(20),T3U(20),T3V(20),T3W(20),T3X(20),T3Y(20),T3Z(20),T30(20),T36
     +(20),T38(20),T39(20),T4S(20),DTMP1(20),DTMP2(20),TFAKE(20)
      DATA R/1716./,XK1/7.86711/,XK2/7.59839/,CV/4290./,DX/.01/
      EPS=1.0E-6
      R1(1)=YO(1)
      R2(1)=YO(2)
      XM1(1)=YO(3)
      XM2(1)=YO(4)
      E1(1)=YO(5)
      E2(1)=YO(6)
      TMP1(1)=YO(7)
      TMP2(1)=YO(8)
      TFAKE(1)=YO(9)
      IND=0
      DO 1 III=1,19
      NIII=III
      IIII=III - 1
      TST(III)=XM2(III)-XM1(III)
      TSU(III)=TST(III)/DX
      TSV(III)=-TSU(III)
      TSW(III)=XM1(III)*XK1
      TSX(III)=TSV(III)-TSW(III)
      DR1(III)=TSX(III)
      R1(III + 1)=DR1(III)/FLOAT(III)
      TS1(III)=XM2(III)*XK2
      TS2(III)=TSV(III)-TS1(III)
      DR2(III)=TS2(III)
      R2(III + 1)=DR2(III)/FLOAT(III)
      TS3(III)=0.
      DO 100 JJJ=1,III
  100 TS3(III)=TS3(III)+XM1(JJJ)*XM1(III-JJJ+1)
      IF (III.EQ.1) GO TO 101
      TS4(III)=TS3(III)-TS4(1)*R1(III)
      IF(III.EQ.2)GO TO 102
      DO 103 JJJ=2,IIII
  103 TS4(III)=TS4(III)-TS4(JJJ)*R1(III-JJJ+1)
  102 TS4(III)=TS4(III)/R1(1)
      GO TO 104
  101 TS4(III)=TS3(III)/R1(III)
  104 CONTINUE
      TS5(III)=TS4(III)*XK1
      TS6(III)=-TS5(III)
      IF (III.EQ.1) GO TO 105
      TS7(III)=XM1(III)-TS7(1)*R1(III)
```

```
      IF(III.EQ.2)GO TO 106
      DO 107 JJJ=2,IIII
107   TS7(III)=TS7(III)-TS7(JJJ)*R1(III-JJJ+1)
106   TS7(III)=TS7(III)/R1(1)
      GO TO 108
105   TS7(III)=XM1(III)/R1(III)
108   CONTINUE
      TS8(III)=0.
      DO 109 JJJ=1,III
109   TS8(III)=TS8(III)+TS7(JJJ)*TS7(III-JJJ+1)
      TS9(III)=TMP1(III)*R
      TTS(III)=TS8(III)-TS9(III)
      TTT(III)=R2(III)-R1(III)
      TTU(III)=0.
      DO 110 JJJ=1,III
110   TTU(III)=TTU(III)+TTS(JJJ)*TTT(III-JJJ+1)
      TTV(III)=TTU(III)/DX
      TTW(III)=TS6(III)+TTV(III)
      TTX(III)=XM1(III)*2.
      IF (III.EQ.1) GO TO 111
      TTY(III)=TTX(III)-TTY(1)*R1(III)
      IF(III.EQ.2)GO TO 112
      DO 113 JJJ=2,IIII
113   TTY(III)=TTY(III)-TTY(JJJ)*R1(III-JJJ+1)
112   TTY(III)=TTY(III)/R1(1)
      GO TO 114
111   TTY(III)=TTX(III)/R1(III)
114   CONTINUE
      TTO(III)=0.
      DO 115 JJJ=1,III
115   TTO(III)=TTO(III)+TTY(JJJ)*TST(III-JJJ+1)
      TT1(III)=TTO(III)/DX
      TT2(III)=TTW(III)-TT1(III)
      DXM1(III)=TT2(III)
      XM1(III + 1)=DXM1(III)/FLOAT(III)
      TT3(III)=0.
      DO 116 JJJ=1,III
116   TT3(III)=TT3(III)+XM2(JJJ)*XM2(III-JJJ+1)
      IF (III.EQ.1) GO TO 117
      TT4(III)=TT3(III)-TT4(1)*R2(III)
      IF(III.EQ.2)GO TO 118
      DO 119 JJJ=2,IIII
119   TT4(III)=TT4(III)-TT4(JJJ)*R2(III-JJJ+1)
118   TT4(III)=TT4(III)/R2(1)
      GO TO 120
117   TT4(III)=TT3(III)/R2(III)
120   CONTINUE
      TT5(III)=TT4(III)*XK2
      TT6(III)=-TT5(III)
      IF (III.EQ.1) GO TO 121
      TT7(III)=XM2(III)-TT7(1)*R2(III)
      IF(III.EQ.2)GO TO 122
      DO 123 JJJ=2,IIII
123   TT7(III)=TT7(III)-TT7(JJJ)*R2(III-JJJ+1)
122   TT7(III)=TT7(III)/R2(1)
      GO TO 124
121   TT7(III)=XM2(III)/R2(III)
124   CONTINUE
```

```
      TT8(III)=0.
      DO 125 JJJ=1,III
  125 TT8(III)=TT8(III)+TT7(JJJ)*TT7(III-JJJ+1)
      TT9(III)=TMP2(III)*R
      TUS(III)=TT8(III)-TT9(III)
      TUU(III)=0.
      DO 126 JJJ=1,III
  126 TUU(III)=TUU(III)+TUS(JJJ)*TTT(III-JJJ+1)
      TUV(III)=TUU(III)/DX
      TUW(III)=TT6(III)+TUV(III)
      TUY(III)=TT7(III)*2.
      TU0(III)=0.
      DO 127 JJJ=1,III
  127 TU0(III)=TU0(III)+TUY(JJJ)*TST(III-JJJ+1)
      TU1(III)=TU0(III)/DX
      TU2(III)=TUW(III)-TU1(III)
      DXM2(III)=TU2(III)
      XM2(III + 1)=DXM2(III)/FLOAT(III)
      IF (III.EQ.1) GO TO 128
      TU3(III)=E1(III)-TU3(1)*R1(III)
      IF(III.EQ.2)GO TO 129
      DO 130 JJJ=2,IIII
  130 TU3(III)=TU3(III)-TU3(JJJ)*R1(III-JJJ+1)
  129 TU3(III)=TU3(III)/R1(1)
      GO TO 131
  128 TU3(III)=E1(III)/R1(III)
  131 CONTINUE
      TU5(III)=TU3(III)+TS9(III)
      TU6(III)=0.
      DO 132 JJJ=1,III
  132 TU6(III)=TU6(III)+XM1(JJJ)*TU5(III-JJJ+1)
      TU7(III)=TU6(III)*XK1
      TU8(III)=-TU7(III)
      TU9(III)=0.
      DO 133 JJJ=1,III
  133 TU9(III)=TU9(III)+XM1(JJJ)*E1(III-JJJ+1)
      TVS(III)=0.
      DO 134 JJJ=1,III
  134 TVS(III)=TVS(III)+R1(JJJ)*R1(III-JJJ+1)
      IF (III.EQ.1) GO TO 135
      TVT(III)=TU9(III)-TVT(1)*TVS(III)
      IF(III.EQ.2)GO TO 136
      DO 137 JJJ=2,IIII
  137 TVT(III)=TVT(III)-TVT(JJJ)*TVS(III-JJJ+1)
  136 TVT(III)=TVT(III)/TVS(1)
      GO TO 138
  135 TVT(III)=TU9(III)/TVS(III)
  138 CONTINUE
      TVV(III)=0.
      DO 139 JJJ=1,III
  139 TVV(III)=TVV(III)+TVT(JJJ)*TTT(III-JJJ+1)
      TVW(III)=TVV(III)/DX
      TVX(III)=TU8(III)+TVW(III)
      TV2(III)=0.
      DO 140 JJJ=1,III
  140 TV2(III)=TV2(III)+TU5(JJJ)*TST(III-JJJ+1)
      TV3(III)=TV2(III)/DX
      TV4(III)=TVX(III)-TV3(III)
```

```
      TV6(III)=E2(III)-E1(III)
      TV7(III)=0.
      DO 141 JJJ=1,III
141   TV7(III)=TV7(III)+TS7(JJJ)*TV6(III-JJJ+1)
      TV8(III)=TV7(III)/DX
      TV9(III)=TV4(III)-TV8(III)
      DE1(III)=TV9(III)
      E1(III + 1)=DE1(III)/FLOAT(III)
      IF (III.EQ.1) GO TO 142
      TWS(III)=E2(III)-TWS(1)*R2(III)
      IF(III.EQ.2)GO TO 143
      DO 144 JJJ=2,IIII
144   TWS(III)=TWS(III)-TWS(JJJ)*R2(III-JJJ+1)
143   TWS(III)=TWS(III)/R2(1)
      GO TO 145
142   TWS(III)=E2(III)/R2(III)
145   CONTINUE
      TWU(III)=TWS(III)+TT9(III)
      TWV(III)=0.
      DO 146 JJJ=1,III
146   TWV(III)=TWV(III)+XM2(JJJ)*TWU(III-JJJ+1)
      TWW(III)=TWV(III)*XK2
      TWX(III)=-TWW(III)
      TWY(III)=0.
      DO 147 JJJ=1,III
147   TWY(III)=TWY(III)+XM2(JJJ)*E2(III-JJJ+1)
      TWZ(III)=0.
      DO 148 JJJ=1,III
148   TWZ(III)=TWZ(III)+R2(JJJ)*R2(III-JJJ+1)
      IF (III.EQ.1) GO TO 149
      TWO(III)=TWY(III)-TWO(1)*TWZ(III)
      IF(III.EQ.2)GO TO 150
      DO 151 JJJ=2,IIII
151   TWO(III)=TWO(III)-TWO(JJJ)*TWZ(III-JJJ+1)
150   TWO(III)=TWO(III)/TWZ(1)
      GO TO 152
149   TWO(III)=TWY(III)/TWZ(III)
152   CONTINUE
      TW2(III)=0.
      DO 153 JJJ=1,III
153   TW2(III)=TW2(III)+TWO(JJJ)*TTT(III-JJJ+1)
      TW3(III)=TW2(III)/DX
      TW4(III)=TWX(III)+TW3(III)
      TW9(III)=0.
      DO 154 JJJ=1,III
154   TW9(III)=TW9(III)+TWU(JJJ)*TST(III-JJJ+1)
      TXS(III)=TW9(III)/DX
      TXT(III)=TW4(III)-TXS(III)
      TXW(III)=0.
      DO 155 JJJ=1,III
155   TXW(III)=TXW(III)+TT7(JJJ)*TV6(III-JJJ+1)
      TXX(III)=TXW(III)/DX
      TXY(III)=TXT(III)-TXX(III)
      DE2(III)=TXY(III)
      E2(III + 1)=DE2(III)/FLOAT(III)
      TZ6(III)=0.
      DO 156 JJJ=1,III
156   TZ6(III)=TZ6(III)+XM1(JJJ)*TT2(III-JJJ+1)
```

```
      IF (III.EQ.1) GO TO 157
      TZ7(III)=TZ6(III)-TZ7(1)*R1(III)
      IF(III.EQ.2)GO TO 158
      DO 159 JJJ=2,IIII
159   TZ7(III)=TZ7(III)-TZ7(JJJ)*R1(III-JJJ+1)
158   TZ7(III)=TZ7(III)/R1(1)
      GO TO 160
157   TZ7(III)=TZ6(III)/R1(III)
160   CONTINUE
      TZ8(III)=TV9(III)-TZ7(III)
      IF (III.EQ.1) GO TO 161
      TZ9(III)=TZ8(III)-TZ9(1)*R1(III)
      IF(III.EQ.2)GO TO 162
      DO 163 JJJ=2,IIII
163   TZ9(III)=TZ9(III)-TZ9(JJJ)*R1(III-JJJ+1)
162   TZ9(III)=TZ9(III)/R1(1)
      GO TO 164
161   TZ9(III)=TZ8(III)/R1(III)
164   CONTINUE
      TOS(III)=0.
      DO 165 JJJ=1,III
165   TOS(III)=TOS(III)+XM1(JJJ)*XM1(III-JJJ+1)
      IF (III.EQ.1) GO TO 166
      TOT(III)=TOS(III)-TOT(1)*R1(III)
      IF(III.EQ.2)GO TO 167
      DO 168 JJJ=2,IIII
168   TOT(III)=TOT(III)-TOT(JJJ)*R1(III-JJJ+1)
167   TOT(III)=TOT(III)/R1(1)
      GO TO 169
166   TOT(III)=TOS(III)/R1(III)
169   CONTINUE
      TOU(III)=E1(III)-TOT(III)
      TOO(III)=0.
      DO 170 JJJ=1,III
170   TOO(III)=TOO(III)+TOU(JJJ)*TSX(III-JJJ+1)
      IF (III.EQ.1) GO TO 171
      TO2(III)=TOO(III)-TO2(1)*TVS(III)
      IF(III.EQ.2)GO TO 172
      DO 173 JJJ=2,IIII
173   TO2(III)=TO2(III)-TO2(JJJ)*TVS(III-JJJ+1)
172   TO2(III)=TO2(III)/TVS(1)
      GO TO 174
171   TO2(III)=TOO(III)/TVS(III)
174   CONTINUE
      TO3(III)=TZ9(III)-TO2(III)
      TO4(III)=TO3(III)/CV
      DTMP1(III)=TO4(III)
      TMP1(III + 1)=DTMP1(III)/FLOAT(III)
      T3U(III)=0.
      DO 175 JJJ=1,III
175   T3U(III)=T3U(III)+XM2(JJJ)*TU2(III-JJJ+1)
      IF (III.EQ.1) GO TO 176
      T3V(III)=T3U(III)-T3V(1)*R2(III)
      IF(III.EQ.2)GO TO 177
      DO 178 JJJ=2,IIII
178   T3V(III)=T3V(III)-T3V(JJJ)*R2(III-JJJ+1)
177   T3V(III)=T3V(III)/R2(1)
      GO TO 179
```

```
 178   T3V(III)=T3U(III)/R2(III)
 179   CONTINUE
       T3W(III)=TXY(III)-T3V(III)
       IF (III.EQ.1) GO TO 180
       T3X(III)=T3W(III)-T3X(1)*R2(III)
       IF(III.EQ.2)GO TO 181
       DO 182 JJJ=2,IIII
 182   T3X(III)=T3X(III)-T3X(JJJ)*R2(III-JJJ+1)
 181   T3X(III)=T3X(III)/R2(1)
       GO TO 183
 180   T3X(III)=T3W(III)/R2(III)
 183   CONTINUE
       T3Y(III)=0.
       DO 184 JJJ=1,III
 184   T3Y(III)=T3Y(III)+XM2(JJJ)*XM2(III-JJJ+1)
       IF (III.EQ.1) GO TO 185
       T3Z(III)=T3Y(III)-T3Z(1)*R2(III)
       IF(III.EQ.2)GO TO 186
       DO 187 JJJ=2,IIII
 187   T3Z(III)=T3Z(III)-T3Z(JJJ)*R2(III-JJJ+1)
 186   T3Z(III)=T3Z(III)/R2(1)
       GO TO 188
 185   T3Z(III)=T3Y(III)/R2(III)
 188   CONTINUE
       T30(III)=E2(III)-T3Z(III)
       T36(III)=0.
       DO 189 JJJ=1,III
 189   T36(III)=T36(III)+T30(JJJ)*TS2(III-JJJ+1)
       IF (III.EQ.1) GO TO 190
       T38(III)=T36(III)-T38(1)*TWZ(III)
       IF(III.EQ.2)GO TO 191
       DO 192 JJJ=2,IIII
 192   T38(III)=T38(III)-T38(JJJ)*TWZ(III-JJJ+1)
 191   T38(III)=T38(III)/TWZ(1)
       GO TO 193
 190   T38(III)=T36(III)/TWZ(III)
 193   CONTINUE
       T39(III)=T3X(III)-T38(III)
       T4S(III)=T39(III)/CV
       DTMP2(III)=T4S(III)
       TMP2(III + 1)=DTMP2(III)/FLOAT(III)
       IF (III.GT.1)DTFAKE(III)=0.
       DTFAKE(1)=1.
       TFAKE(III + 1)=DTFAKE(III)/FLOAT(III)
       IF (III.LT.4)GO TO 1
       IIII=III + 1
       ZZZZ1=0.
       ZZZZ2=0.
       DO 194 JJJ=1,IIII
       ZZZZ1=ZZZZ1 +R1(JJJ)
       IF(JJJ.LT.III-4)GO TO 194
       ZZZZ2=ZZZZ2 + ABS(R1(JJJ))
 194   CONTINUE
       ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
       IF(ZZZZ2.GT.ZZZZ1)GO TO 1
       ZZZZ1=0.
       ZZZZ2=0.
       DO 195 JJJ=1,IIII
```

```
      ZZZZ1=ZZZZ1 +R2(JJJ)
      IF(JJJ.LT.III-4)GO TO 195
      ZZZZ2=ZZZZ2 + ABS(R2(JJJ))
195   CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 196 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +XM1(JJJ)
      IF(JJJ.LT.III-4)GO TO 196
      ZZZZ2=ZZZZ2 + ABS(XM1(JJJ))
196   CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 197 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +XM2(JJJ)
      IF(JJJ.LT.III-4)GO TO 197
      ZZZZ2=ZZZZ2 + ABS(XM2(JJJ))
197   CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 198 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +E1(JJJ)
      IF(JJJ.LT.III-4)GO TO 198
      ZZZZ2=ZZZZ2 + ABS(E1(JJJ))
198   CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 199 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +E2(JJJ)
      IF(JJJ.LT.III-4)GO TO 199
      ZZZZ2=ZZZZ2 + ABS(E2(JJJ))
199   CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 200 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +TMP1(JJJ)
      IF(JJJ.LT.III-4)GO TO 200
      ZZZZ2=ZZZZ2 + ABS(TMP1(JJJ))
200   CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 201 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +TMP2(JJJ)
      IF(JJJ.LT.III-4)GO TO 201
      ZZZZ2=ZZZZ2 + ABS(TMP2(JJJ))
201   CONTINUE
```

```
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 202 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +TFAKE(JJJ)
      IF(JJJ.LT.III-4)GO TO 202
      ZZZZ2=ZZZZ2 + ABS(TFAKE(JJJ))
202   CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      GO TO 2
    1 CONTINUE
    2 CONTINUE
      DO 203 JJJ=1,NIII
      IF(ABS(R1(JJJ)).LT.EPS) GO TO 203
      KKK=JJJ
      GO TO 204
203   CONTINUE
204   ZZZZ1=0.
      KKK1=KKK + 1
      DO 205 JJJ=KKK1,NIII
205   ZZZZ1=ZZZZ1+ABS(R1(JJJ))
      IF(ZZZZ1/ABS(R1(KKK)).GE.1)IND=IND + 1
      DO 206 JJJ=1,NIII
      IF(ABS(R1(JJJ)).LT.EPS) GO TO 206
      KKK=JJJ
      GO TO 207
206   CONTINUE
207   ZZZZ1=0.
      KKK1=KKK + 1
      DO 208 JJJ=KKK1,NIII
208   ZZZZ1=ZZZZ1+ABS(R1(JJJ))
      IF(ZZZZ1/ABS(R1(KKK)).GE.1)IND=IND + 1
      DO 209 JJJ=1,NIII
      IF(ABS(R1(JJJ)).LT.EPS) GO TO 209
      KKK=JJJ
      GO TO 210
209   CONTINUE
210   ZZZZ1=0.
      KKK1=KKK + 1
      DO 211 JJJ=KKK1,NIII
211   ZZZZ1=ZZZZ1+ABS(R1(JJJ))
      IF(ZZZZ1/ABS(R1(KKK)).GE.1)IND=IND + 1
      DO 212 JJJ=1,NIII
      IF(ABS(R2(JJJ)).LT.EPS) GO TO 212
      KKK=JJJ
      GO TO 213
212   CONTINUE
213   ZZZZ1=0.
      KKK1=KKK + 1
      DO 214 JJJ=KKK1,NIII
214   ZZZZ1=ZZZZ1+ABS(R2(JJJ))
      IF(ZZZZ1/ABS(R2(KKK)).GE.1)IND=IND + 1
      DO 215 JJJ=1,NIII
      IF(ABS(R2(JJJ)).LT.EPS) GO TO 215
      KKK=JJJ
      GO TO 216
```

```
  215   CONTINUE
  216   ZZZZ1=0.
        KKK1=KKK + 1
        DO 217 JJJ=KKK1,NIII
  217   ZZZZ1=ZZZZ1+ABS(R2(JJJ))
        IF(ZZZZ1/ABS(R2(KKK)).GE.1)IND=IND + 1
        DO 218 JJJ=1,NIII
        IF(ABS(R1(JJJ)).LT.EPS) GO TO 218
        KKK=JJJ
        GO TO 219
  218   CONTINUE
  219   ZZZZ1=0.
        KKK1=KKK + 1
        DO 220 JJJ=KKK1,NIII
  220   ZZZZ1=ZZZZ1+ABS(R1(JJJ))
        IF(ZZZZ1/ABS(R1(KKK)).GE.1)IND=IND + 1
        DO 221 JJJ=1,NIII
        IF(ABS(TVS(JJJ)).LT.EPS) GO TO 221
        KKK=JJJ
        GO TO 222
  221   CONTINUE
  222   ZZZZ1=0.
        KKK1=KKK + 1
        DO 223 JJJ=KKK1,NIII
  223   ZZZZ1=ZZZZ1+ABS(TVS(JJJ))
        IF(ZZZZ1/ABS(TVS(KKK)).GE.1)IND=IND + 1
        DO 224 JJJ=1,NIII
        IF(ABS(R2(JJJ)).LT.EPS) GO TO 224
        KKK=JJJ
        GO TO 225
  224   CONTINUE
  225   ZZZZ1=0.
        KKK1=KKK + 1
        DO 226 JJJ=KKK1,NIII
  226   ZZZZ1=ZZZZ1+ABS(R2(JJJ))
        IF(ZZZZ1/ABS(R2(KKK)).GE.1)IND=IND + 1
        DO 227 JJJ=1,NIII
        IF(ABS(TWZ(JJJ)).LT.EPS) GO TO 227
        KKK=JJJ
        GO TO 228
  227   CONTINUE
  228   ZZZZ1=0.
        KKK1=KKK + 1
        DO 229 JJJ=KKK1,NIII
  229   ZZZZ1=ZZZZ1+ABS(TWZ(JJJ))
        IF(ZZZZ1/ABS(TWZ(KKK)).GE.1)IND=IND + 1
        DO 230 JJJ=1,NIII
        IF(ABS(R1(JJJ)).LT.EPS) GO TO 230
        KKK=JJJ
        GO TO 231
  230   CONTINUE
  231   ZZZZ1=0.
        KKK1=KKK + 1
        DO 232 JJJ=KKK1,NIII
  232   ZZZZ1=ZZZZ1+ABS(R1(JJJ))
        IF(ZZZZ1/ABS(R1(KKK)).GE.1)IND=IND + 1
        DO 233 JJJ=1,NIII
        IF(ABS(R1(JJJ)).LT.EPS) GO TO 233
```

```
      KKK=JJJ
      GO TO 234
233   CONTINUE
234   ZZZZ1=0.
      KKK1=KKK + 1
      DO 235 JJJ=KKK1,NIII
235   ZZZZ1=ZZZZ1+ABS(R1(JJJ))
      IF(ZZZZ1/ABS(R1(KKK)).GE.1)IND=IND + 1
      DO 236 JJJ=1,NIII
      IF(ABS(R1(JJJ)).LT.EPS) GO TO 236
      KKK=JJJ
      GO TO 237
236   CONTINUE
237   ZZZZ1=0.
      KKK1=KKK + 1
      DO 238 JJJ=KKK1,NIII
238   ZZZZ1=ZZZZ1+ABS(R1(JJJ))
      IF(ZZZZ1/ABS(R1(KKK)).GE.1)IND=IND + 1
      DO 239 JJJ=1,NIII
      IF(ABS(TVS(JJJ)).LT.EPS) GO TO 239
      KKK=JJJ
      GO TO 240
239   CONTINUE
240   ZZZZ1=0.
      KKK1=KKK + 1
      DO 241 JJJ=KKK1,NIII
241   ZZZZ1=ZZZZ1+ABS(TVS(JJJ))
      IF(ZZZZ1/ABS(TVS(KKK)).GE.1)IND=IND + 1
      DO 242 JJJ=1,NIII
      IF(ABS(R2(JJJ)).LT.EPS) GO TO 242
      KKK=JJJ
      GO TO 243
242   CONTINUE
243   ZZZZ1=0.
      KKK1=KKK + 1
      DO 244 JJJ=KKK1,NIII
244   ZZZZ1=ZZZZ1+ABS(R2(JJJ))
      IF(ZZZZ1/ABS(R2(KKK)).GE.1)IND=IND + 1
      DO 245 JJJ=1,NIII
      IF(ABS(R2(JJJ)).LT.EPS) GO TO 245
      KKK=JJJ
      GO TO 246
245   CONTINUE
246   ZZZZ1=0.
      KKK1=KKK + 1
      DO 247 JJJ=KKK1,NIII
247   ZZZZ1=ZZZZ1+ABS(R2(JJJ))
      IF(ZZZZ1/ABS(R2(KKK)).GE.1)IND=IND + 1
      DO 248 JJJ=1,NIII
      IF(ABS(R2(JJJ)).LT.EPS) GO TO 248
      KKK=JJJ
      GO TO 249
248   CONTINUE
249   ZZZZ1=0.
      KKK1=KKK + 1
      DO 250 JJJ=KKK1,NIII
250   ZZZZ1=ZZZZ1+ABS(R2(JJJ))
      IF(ZZZZ1/ABS(R2(KKK)).GE.1)IND=IND + 1
```

```
      DO 251 JJJ=1,NIII
      IF(ABS(TWZ(JJJ)).LT.EPS) GO TO 251
      KKK=JJJ
      GO TO 252
251   CONTINUE
252   ZZZZ1=0.
      KKK1=KKK + 1
      DO 253 JJJ=KKK1,NIII
253   ZZZZ1=ZZZZ1+ABS(TWZ(JJJ))
      IF(ZZZZ1/ABS(TWZ(KKK)).GE.1)IND=IND + 1
      NIII=NIII + 1
      ZZZB(1)=R1(NIII)
      DO 254JJJ=2,NIII
254   ZZZB(JJJ)=R1(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(1)=ZZZB(NIII)
      ZZZB(1)=R2(NIII)
      DO 255JJJ=2,NIII
255   ZZZB(JJJ)=R2(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(2)=ZZZB(NIII)
      ZZZB(1)=XM1(NIII)
      DO 256JJJ=2,NIII
256   ZZZB(JJJ)=XM1(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(3)=ZZZB(NIII)
      ZZZB(1)=XM2(NIII)
      DO 257JJJ=2,NIII
257   ZZZB(JJJ)=XM2(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(4)=ZZZB(NIII)
      ZZZB(1)=E1(NIII)
      DO 258JJJ=2,NIII
258   ZZZB(JJJ)=E1(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(5)=ZZZB(NIII)
      ZZZB(1)=E2(NIII)
      DO 259JJJ=2,NIII
259   ZZZB(JJJ)=E2(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(6)=ZZZB(NIII)
      ZZZB(1)=TMP1(NIII)
      DO 260JJJ=2,NIII
260   ZZZB(JJJ)=TMP1(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(7)=ZZZB(NIII)
      ZZZB(1)=TMP2(NIII)
      DO 261JJJ=2,NIII
261   ZZZB(JJJ)=TMP2(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(8)=ZZZB(NIII)
      ZZZB(1)=TFAKE(NIII)
      DO 262JJJ=2,NIII
262   ZZZB(JJJ)=TFAKE(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(9)=ZZZB(NIII)
      RETURN
      END
```

APPENDIX B

Output of SERIES for constant spacial derivative formulation

```
      SUBROUTINE DIFFUN(T,Y,DY)
      DIMENSION DY(20),Y(20)
      DATA R/1716./,DX/.01/
      DY(1)=0.
      DY(2)=-Y(5)*Y(12)-(Y(5)-Y(4))/DX
      DY(3)=-Y(6)*Y(13)-(Y(6)-Y(5))/DX
      DY(4)=0.
      DY(5)=-Y(5)*Y(5)/Y(2)*Y(12)-(R*Y(10)-(Y(5)/Y(2))**2)*(Y(2)-Y(1))/
     +DX-2.*Y(5)/Y(2)*(Y(5)-Y(4))/DX
      DY(6)=-Y(6)*Y(6)/Y(3)*Y(13)-(R*Y(11)-(Y(6)/Y(3))**2)*(Y(3)-Y(2))/
     +DX-2.*Y(6)/Y(3)*(Y(6)-Y(5))/DX
      DY(7)=0.
      DY(8)=-Y(5)*(Y(8)/Y(2)+R*Y(10))*Y(12)+Y(5)*Y(8)/Y(2)**2*(Y(2)-Y(1
     +))/DX-(Y(8)/Y(2)+R*Y(10))*(Y(5)-Y(4))/DX-Y(5)/Y(2)*(Y(8)-Y(7))/DX
      DY(9)=-Y(6)*(Y(9)/Y(3)+R*Y(11))*Y(13)+Y(6)*Y(9)/Y(3)**2*(Y(3)-Y(2
     +))/DX-(Y(9)/Y(3)+R*Y(11))*(Y(6)-Y(5))/DX-Y(6)/Y(3)*(Y(9)-Y(8))/DX
      DY(10)=0.
      DY(11)=0.
      DY(12)=0.
      DY(13)=0.
      RETURN
      END
```

```
      SUBROUTINE SOL(T,YO,YNEW,IND)
      DIMENSION YO(20),YNEW(20),ZZZB(20),DRO(20),RO(20),DR1(20),R1(20),
     +XM1(20),XK1(20),TST(20),TSU(20),XMO(20),TSV(20),TSW(20),TSX(20),D
     +R2(20),R2(20),XM2(20),XK2(20),TSY(20),TSZ(20),TSO(20),TS1(20),TS2
     +(20),DXMO(20),DXM1(20),TS3(20),TS4(20),TS5(20),TS6(20),TMP1(20),T
     +S7(20),TS8(20),TS9(20),TTS(20),TTT(20),TTU(20),TTV(20),TTW(20),TT
     +X(20),TTY(20),TTO(20),TT1(20),TT2(20),DXM2(20),TT3(20),TT4(20),TT
     +5(20),TT6(20),TMP2(20),TT7(20),TT8(20),TT9(20),TUS(20),TUT(20),TU
     +U(20),TUV(20),TUW(20),TUX(20),TUY(20),TUO(20),TU1(20),TU2(20),DEO
     +(20),EO(20),DE1(20),E1(20),TU3(20),TU5(20),TU6(20),TU7(20),TU8
     +(20),TU9(20),TVS(20),TVT(20),TVV(20),TVW(20),TVX(20),TV2(20),TV3
     +(20),TV4(20),TV6(20),TV7(20),TV8(20),TV9(20),DE2(20),E2(20),TWS
     +(20),TWU(20),TWV(20),TWW(20),TWX(20),TWY(20),TWZ(20),TWO(20),TW2
     +(20),TW3(20),TW4(20),TW9(20),TXS(20),TXT(20),TXV(20),TXW(20),TXX
     +(20),TXY(20),DXK1(20),DXK2(20),DTMP1(20),DTMP2(20)
      DATA R/1716./,DX/.01/
      EPS=1.0E-6
      RO(1)=YO(1)
      R1(1)=YO(2)
      R2(1)=YO(3)
      XMO(1)=YO(4)
      XM1(1)=YO(5)
      XM2(1)=YO(6)
      EO(1)=YO(7)
      E1(1)=YO(8)
      E2(1)=YO(9)
      TMP1(1)=YO(10)
      TMP2(1)=YO(11)
      XK1(1)=YO(12)
      XK2(1)=YO(13)
      IND=0
      DO 1 III=1,19
      NIII=III
      IIII=III - 1
      IF (III.GT.1)DRO(III)=0.
      DRO(1)=0.
      RO(III + 1)=DRO(III)/FLOAT(III)
      TST(III)=0.
      DO 100 JJJ=1,III
  100 TST(III)=TST(III)+XM1(JJJ)*XK1(III-JJJ+1)
      TSU(III)=-TST(III)
      TSV(III)=XM1(III)-XMO(III)
      TSW(III)=TSV(III)/DX
      TSX(III)=TSU(III)-TSW(III)
      DR1(III)=TSX(III)
      R1(III + 1)=DR1(III)/FLOAT(III)
      TSY(III)=0.
      DO 101 JJJ=1,III
  101 TSY(III)=TSY(III)+XM2(JJJ)*XK2(III-JJJ+1)
      TSZ(III)=-TSY(III)
      TSO(III)=XM2(III)-XM1(III)
      TS1(III)=TSO(III)/DX
      TS2(III)=TSZ(III)-TS1(III)
      DR2(III)=TS2(III)
      R2(III + 1)=DR2(III)/FLOAT(III)
      IF (III.GT.1)DXMO(III)=0.
      DXMO(1)=0.
      XMO(III + 1)=DXMO(III)/FLOAT(III)
```

```
      TS3(III)=C.
      DC 102 JJJ=1,III
102   TS3(III)=TS3(III)+XM1(JJJ)*XM1(III-JJJ+1)
      IF (III.EC.1) GC TO 103
      TS4(III)=TS3(III)-TS4(1)*R1(III)
      IF(III.EQ.2)GC TO 104
      DC 105 JJJ=2,IIII
105   TS4(III)=TS4(III)-TS4(JJJ)*R1(III-JJJ+1)
104   TS4(III)=TS4(III)/R1(1)
      GC TC 106
103   TS4(III)=TS3(III)/R1(III)
106   CCNTINUE
      TS5(III)=0.
      DC 107 JJJ=1,III
107   TS5(III)=TS5(III)+TS4(JJJ)*XK1(III-JJJ+1)
      TS6(III)=-TS5(III)
      TS7(III)=TMP1(III)*R
      IF (III.EC.1) GC TC 108
      TS8(III)=XM1(III)-TS8(1)*R1(III)
      IF(III.EC.2)GC TO 109
      DC 110 JJJ=2,IIII
110   TS8(III)=TS8(III)-TS8(JJJ)*R1(III-JJJ+1)
109   TS8(III)=TS8(III)/R1(1)
      GC TC 111
108   TS8(III)=XM1(III)/R1(III)
111   CCNTINUE
      TS9(III)=C.
      DC 112 JJJ=1,III
112   TS9(III)=TS9(III)+TS8(JJJ)*TS8(III-JJJ+1)
      TTS(III)=TS7(III)-TS9(III)
      TTT(III)=R1(III)-R0(III)
      TTU(III)=0.
      DC 113 JJJ=1,III
113   TTU(III)=TTU(III)+TTS(JJJ)*TTT(III-JJJ+1)
      TTV(III)=TTU(III)/CX
      TTW(III)=TS6(III)-TTV(III)
      TTX(III)=XM1(III)*2.
      IF (III.EC.1) GC TC 114
      TTY(III)=TTX(III)-TTY(1)*R1(III)
      IF(III.EC.2)GC TO 115
      DC 116 JJJ=2,IIII
116   TTY(III)=TTY(III)-TTY(JJJ)*R1(III-JJJ+1)
115   TTY(III)=TTY(III)/R1(1)
      GC TC 117
114   TTY(III)=TTX(III)/R1(III)
117   CCNTINUE
      TT0(III)=0.
      DC 118 JJJ=1,III
118   TT0(III)=TT0(III)+TTY(JJJ)*TSV(III-JJJ+1)
      TT1(III)=TT0(III)/CX
      TT2(III)=TTW(III)-TT1(III)
      DXM1(III)=TT2(III)
      XM1(III + 1)=DXM1(III)/FLCAT(III)
      TT3(III)=C.
      DC 119 JJJ=1,III
119   TT3(III)=TT3(III)+XM2(JJJ)*XM2(III-JJJ+1)
      IF (III.EC.1) GC TC 120
      TT4(III)=TT3(III)-TT4(1)*R2(III)
```

```
      IF(III.EQ.2)GO TO 121
      DO 122 JJJ=2,IIII
122   TT4(III)=TT4(III)-TT4(JJJ)*R2(III-JJJ+1)
121   TT4(III)=TT4(III)/R2(1)
      GO TO 123
120   TT4(III)=TT3(III)/R2(III)
123   CONTINUE
      TT5(III)=0.
      DO 124 JJJ=1,III
124   TT5(III)=TT5(III)+TT4(JJJ)*XK2(III-JJJ+1)
      TT6(III)=-TT5(III)
      TT7(III)=TMP2(III)*R
      IF (III.EQ.1) GO TO 125
      TT8(III)=XM2(III)-TT8(1)*R2(III)
      IF(III.EQ.2)GO TO 126
      DO 127 JJJ=2,IIII
127   TT8(III)=TT8(III)-TT8(JJJ)*R2(III-JJJ+1)
126   TT8(III)=TT8(III)/R2(1)
      GO TO 128
125   TT8(III)=XM2(III)/R2(III)
128   CONTINUE
      TT9(III)=0.
      DO 129 JJJ=1,III
129   TT9(III)=TT9(III)+TT8(JJJ)*TT8(III-JJJ+1)
      TUS(III)=TT7(III)-TT9(III)
      TUT(III)=R2(III)-R1(III)
      TUU(III)=0.
      DO 130 JJJ=1,III
130   TUU(III)=TUU(III)+TUS(JJJ)*TUT(III-JJJ+1)
      TUV(III)=TUU(III)/DX
      TUW(III)=TT6(III)-TUV(III)
      TUX(III)=XM2(III)*2.
      IF (III.EQ.1) GO TO 131
      TUY(III)=TUX(III)-TUY(1)*R2(III)
      IF(III.EQ.2)GO TO 132
      DO 133 JJJ=2,IIII
133   TUY(III)=TUY(III)-TUY(JJJ)*R2(III-JJJ+1)
132   TUY(III)=TUY(III)/R2(1)
      GO TO 134
131   TUY(III)=TUX(III)/R2(III)
134   CONTINUE
      TU0(III)=0.
      DO 135 JJJ=1,III
135   TU0(III)=TU0(III)+TUY(JJJ)*TS0(III-JJJ+1)
      TU1(III)=TU0(III)/DX
      TU2(III)=TUW(III)-TU1(III)
      DXM2(III)=TU2(III)
      XM2(III + 1)=DXM2(III)/FLOAT(III)
      IF (III.GT.1)DEO(III)=0.
      DEO(1)=0.
      EO(III + 1)=DEO(III)/FLOAT(III)
      IF (III.EQ.1) GO TO 136
      TU3(III)=E1(III)-TU3(1)*R1(III)
      IF(III.EQ.2)GO TO 137
      DO 138 JJJ=2,IIII
138   TU3(III)=TU3(III)-TU3(JJJ)*R1(III-JJJ+1)
137   TU3(III)=TU3(III)/R1(1)
      GO TO 139
```

```
136   TU3(III)=E1(III)/R1(III)
139   CONTINUE
      TU5(III)=TU3(III)+TS7(III)
      TU6(III)=0.
      DO 140 JJJ=1,III
140   TU6(III)=TU6(III)+XM1(JJJ)*TU5(III-JJJ+1)
      TU7(III)=0.
      DO 141 JJJ=1,III
141   TU7(III)=TU7(III)+TU6(JJJ)*XK1(III-JJJ+1)
      TU8(III)=-TU7(III)
      TU9(III)=0.
      DO 142 JJJ=1,III
142   TU9(III)=TU9(III)+XM1(JJJ)*E1(III-JJJ+1)
      TVS(III)=0.
      DO 143 JJJ=1,III
143   TVS(III)=TVS(III)+R1(JJJ)*R1(III-JJJ+1)
      IF (III.EQ.1) GO TO 144
      TVT(III)=TU9(III)-TVT(1)*TVS(III)
      IF(III.EQ.2)GO TO 145
      DO 146 JJJ=2,IIII
146   TVT(III)=TVT(III)-TVT(JJJ)*TVS(III-JJJ+1)
145   TVT(III)=TVT(III)/TVS(1)
      GO TO 147
144   TVT(III)=TU9(III)/TVS(III)
147   CONTINUE
      TVV(III)=0.
      DO 148 JJJ=1,III
148   TVV(III)=TVV(III)+TVT(JJJ)*TTT(III-JJJ+1)
      TVW(III)=TVV(III)/DX
      TVX(III)=TU8(III)+TVW(III)
      TV2(III)=0.
      DO 149 JJJ=1,III
149   TV2(III)=TV2(III)+TU5(JJJ)*TSV(III-JJJ+1)
      TV3(III)=TV2(III)/DX
      TV4(III)=TVX(III)-TV3(III)
      TV6(III)=E1(III)-E0(III)
      TV7(III)=0.
      DO 150 JJJ=1,III
150   TV7(III)=TV7(III)+TS8(JJJ)*TV6(III-JJJ+1)
      TV8(III)=TV7(III)/DX
      TV9(III)=TV4(III)-TV8(III)
      DE1(III)=TV9(III)
      E1(III + 1)=DE1(III)/FLOAT(III)
      IF (III.EQ.1) GO TO 151
      TWS(III)=E2(III)-TWS(1)*R2(III)
      IF(III.EQ.2)GO TO 152
      DO 153 JJJ=2,IIII
153   TWS(III)=TWS(III)-TWS(JJJ)*R2(III-JJJ+1)
152   TWS(III)=TWS(III)/R2(1)
      GO TO 154
151   TWS(III)=E2(III)/R2(III)
154   CONTINUE
      TWU(III)=TWS(III)+TT7(III)
      TWV(III)=0.
      DO 155 JJJ=1,III
155   TWV(III)=TWV(III)+XM2(JJJ)*TWU(III-JJJ+1)
      TWW(III)=0.
      DO 156 JJJ=1,III
```

```
156   TWW(III)=TWW(III)+TWV(JJJ)*XK2(III-JJJ+1)
      TWX(III)=-TWW(III)
      TWY(III)=0.
      DO 157 JJJ=1,III
157   TWY(III)=TWY(III)+XM2(JJJ)*E2(III-JJJ+1)
      TWZ(III)=0.
      DO 158 JJJ=1,III
158   TWZ(III)=TWZ(III)+R2(JJJ)*R2(III-JJJ+1)
      IF (III.EQ.1) GO TO 159
      TWO(III)=TWY(III)-TWO(1)*TWZ(III)
      IF(III.EQ.2)GO TO 160
      DO 161 JJJ=2,IIII
161   TWO(III)=TWO(III)-TWO(JJJ)*TWZ(III-JJJ+1)
160   TWO(III)=TWO(III)/TWZ(1)
      GO TO 162
159   TWO(III)=TWY(III)/TWZ(III)
162   CONTINUE
      TW2(III)=0.
      DO 163 JJJ=1,III
163   TW2(III)=TW2(III)+TWO(JJJ)*TUT(III-JJJ+1)
      TW3(III)=TW2(III)/DX
      TW4(III)=TWX(III)+TW3(III)
      TW9(III)=0.
      DO 164 JJJ=1,III
164   TW9(III)=TW9(III)+TWU(JJJ)*TSO(III-JJJ+1)
      TXS(III)=TW9(III)/DX
      TXT(III)=TW4(III)-TXS(III)
      TXV(III)=E2(III)-E1(III)
      TXW(III)=0.
      DO 165 JJJ=1,III
165   TXW(III)=TXW(III)+TT8(JJJ)*TXV(III-JJJ+1)
      TXX(III)=TXW(III)/DX
      TXY(III)=TXT(III)-TXX(III)
      DE2(III)=TXY(III)
      E2(III + 1)=DE2(III)/FLOAT(III)
      IF (III.GT.1)DTMP1(III)=0.
      DTMP1(1)=0.
      TMP1(III + 1)=DTMP1(III)/FLOAT(III)
      IF (III.GT.1)DTMP2(III)=0.
      DTMP2(1)=0.
      TMP2(III + 1)=DTMP2(III)/FLOAT(III)
      IF (III.GT.1)DXK1(III)=0.
      DXK1(1)=0.
      XK1(III + 1)=DXK1(III)/FLOAT(III)
      IF (III.GT.1)DXK2(III)=0.
      DXK2(1)=0.
      XK2(III + 1)=DXK2(III)/FLOAT(III)
      IF (III.LT.4)GO TO 1
      IIII=III + 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 166 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +RO(JJJ)
      IF(JJJ.LT.III-4)GO TO 166
      ZZZZ2=ZZZZ2 + ABS(RO(JJJ))
166   CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
```

```
      ZZZZ1=0.
      ZZZZ2=0.
      DO 167 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +R1(JJJ)
      IF(JJJ.LT.III-4)GO TO 167
      ZZZZ2=ZZZZ2 + ABS(R1(JJJ))
  167 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 168 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +R2(JJJ)
      IF(JJJ.LT.III-4)GO TO 168
      ZZZZ2=ZZZZ2 + ABS(R2(JJJ))
  168 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 169 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +XMO(JJJ)
      IF(JJJ.LT.III-4)GO TO 169
      ZZZZ2=ZZZZ2 + ABS(XMO(JJJ))
  169 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 170 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +XM1(JJJ)
      IF(JJJ.LT.III-4)GO TO 170
      ZZZZ2=ZZZZ2 + ABS(XM1(JJJ))
  170 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 171 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +XM2(JJJ)
      IF(JJJ.LT.III-4)GO TO 171
      ZZZZ2=ZZZZ2 + ABS(XM2(JJJ))
  171 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 172 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +E0(JJJ)
      IF(JJJ.LT.III-4)GO TO 172
      ZZZZ2=ZZZZ2 + ABS(E0(JJJ))
  172 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 173 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +E1(JJJ)
```

```
      IF(JJJ.LT.III-4)GO TO 173
      ZZZZ2=ZZZZ2 + ABS(E1(JJJ))
  173 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 174 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +E2(JJJ)
      IF(JJJ.LT.III-4)GO TO 174
      ZZZZ2=ZZZZ2 + ABS(E2(JJJ))
  174 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 175 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +TMP1(JJJ)
      IF(JJJ.LT.III-4)GO TO 175
      ZZZZ2=ZZZZ2 + ABS(TMP1(JJJ))
  175 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 176 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +TMP2(JJJ)
      IF(JJJ.LT.III-4)GO TO 176
      ZZZZ2=ZZZZ2 + ABS(TMP2(JJJ))
  176 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 177 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +XK1(JJJ)
      IF(JJJ.LT.III-4)GO TO 177
      ZZZZ2=ZZZZ2 + ABS(XK1(JJJ))
  177 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      ZZZZ1=0.
      ZZZZ2=0.
      DO 178 JJJ=1,IIII
      ZZZZ1=ZZZZ1 +XK2(JJJ)
      IF(JJJ.LT.III-4)GO TO 178
      ZZZZ2=ZZZZ2 + ABS(XK2(JJJ))
  178 CONTINUE
      ZZZZ1=EPS*(ABS(ZZZZ1) + 1.)
      IF(ZZZZ2.GT.ZZZZ1)GO TO 1
      GO TO 2
    1 CONTINUE
    2 CONTINUE
      DO 179 JJJ=1,NIII
      IF(ABS(R1(JJJ)).LT.EPS) GO TO 179
      KKK=JJJ
      GO TO 180
  179 CONTINUE
```

```
  180   ZZZZ1=0.
        KKK1=KKK + 1
        DO 181 JJJ=KKK1,NIII
  181   ZZZZ1=ZZZZ1+ABS(R1(JJJ))
        IF(ZZZZ1/ABS(R1(KKK)).GE.1)IND=IND + 1
        DO 182 JJJ=1,NIII
        IF(ABS(R1(JJJ)).LT.EPS) GO TO 182
        KKK=JJJ
        GO TO 183
  182   CONTINUE
  183   ZZZZ1=0.
        KKK1=KKK + 1
        DO 184 JJJ=KKK1,NIII
  184   ZZZZ1=ZZZZ1+ABS(R1(JJJ))
        IF(ZZZZ1/ABS(R1(KKK)).GE.1)IND=IND + 1
        DO 185 JJJ=1,NIII
        IF(ABS(R1(JJJ)).LT.EPS) GO TO 185
        KKK=JJJ
        GO TO 186
  185   CONTINUE
  186   ZZZZ1=0.
        KKK1=KKK + 1
        DO 187 JJJ=KKK1,NIII
  187   ZZZZ1=ZZZZ1+ABS(R1(JJJ))
        IF(ZZZZ1/ABS(R1(KKK)).GE.1)IND=IND + 1
        DO 188 JJJ=1,NIII
        IF(ABS(R2(JJJ)).LT.EPS) GO TO 188
        KKK=JJJ
        GO TO 189
  188   CONTINUE
  189   ZZZZ1=0.
        KKK1=KKK + 1
        DO 190 JJJ=KKK1,NIII
  190   ZZZZ1=ZZZZ1+ABS(R2(JJJ))
        IF(ZZZZ1/ABS(R2(KKK)).GE.1)IND=IND + 1
        DO 191 JJJ=1,NIII
        IF(ABS(R2(JJJ)).LT.EPS) GO TO 191
        KKK=JJJ
        GO TO 192
  191   CONTINUE
  192   ZZZZ1=0.
        KKK1=KKK + 1
        DO 193 JJJ=KKK1,NIII
  193   ZZZZ1=ZZZZ1+ABS(R2(JJJ))
        IF(ZZZZ1/ABS(R2(KKK)).GE.1)IND=IND + 1
        DO 194 JJJ=1,NIII
        IF(ABS(R2(JJJ)).LT.EPS) GO TO 194
        KKK=JJJ
        GO TO 195
  194   CONTINUE
  195   ZZZZ1=0.
        KKK1=KKK + 1
        DO 196 JJJ=KKK1,NIII
  196   ZZZZ1=ZZZZ1+ABS(R2(JJJ))
        IF(ZZZZ1/ABS(R2(KKK)).GE.1)IND=IND + 1
        DO 197 JJJ=1,NIII
        IF(ABS(R1(JJJ)).LT.EPS) GO TO 197
        KKK=JJJ
```

```
      GO TO 198
197   CONTINUE
198   ZZZZ1=0.
      KKK1=KKK + 1
      DO 199 JJJ=KKK1,NIII
199   ZZZZ1=ZZZZ1+ABS(R1(JJJ))
      IF(ZZZZ1/ABS(R1(KKK)).GE.1)IND=IND + 1
      DO 200 JJJ=1,NIII
      IF(ABS(TVS(JJJ)).LT.EPS) GO TO 200
      KKK=JJJ
      GO TO 201
200   CONTINUE
201   ZZZZ1=0.
      KKK1=KKK + 1
      DO 202 JJJ=KKK1,NIII
202   ZZZZ1=ZZZZ1+ABS(TVS(JJJ))
      IF(ZZZZ1/ABS(TVS(KKK)).GE.1)IND=IND + 1
      DO 203 JJJ=1,NIII
      IF(ABS(R2(JJJ)).LT.EPS) GO TO 203
      KKK=JJJ
      GO TO 204
203   CONTINUE
204   ZZZZ1=0.
      KKK1=KKK + 1
      DO 205 JJJ=KKK1,NIII
205   ZZZZ1=ZZZZ1+ABS(R2(JJJ))
      IF(ZZZZ1/ABS(R2(KKK)).GE.1)IND=IND + 1
      DO 206 JJJ=1,NIII
      IF(ABS(TWZ(JJJ)).LT.EPS) GO TO 206
      KKK=JJJ
      GO TO 207
206   CONTINUE
207   ZZZZ1=0.
      KKK1=KKK + 1
      DO 208 JJJ=KKK1,NIII
208   ZZZZ1=ZZZZ1+ABS(TWZ(JJJ))
      IF(ZZZZ1/ABS(TWZ(KKK)).GE.1)IND=IND + 1
      NIII=NIII + 1
      ZZZB(1)=RO(NIII)
      DO 209JJJ=2,NIII
209   ZZZB(JJJ)=RO(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(1)=ZZZB(NIII)
      ZZZB(1)=R1(NIII)
      DO 210JJJ=2,NIII
210   ZZZB(JJJ)=R1(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(2)=ZZZB(NIII)
      ZZZB(1)=R2(NIII)
      DO 211JJJ=2,NIII
211   ZZZB(JJJ)=R2(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(3)=ZZZB(NIII)
      ZZZB(1)=XMO(NIII)
      DO 212JJJ=2,NIII
212   ZZZB(JJJ)=XMO(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(4)=ZZZB(NIII)
      ZZZB(1)=XM1(NIII)
      DO 213JJJ=2,NIII
213   ZZZB(JJJ)=XM1(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(5)=ZZZB(NIII)
```

```
      ZZZB(1)=XM2(NIII)
      DO 214JJJ=2,NIII
214   ZZZB(JJJ)=XM2(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(6)=ZZZB(NIII)
      ZZZB(1)=E0(NIII)
      DO 215JJJ=2,NIII
215   ZZZB(JJJ)=E0(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(7)=ZZZB(NIII)
      ZZZB(1)=E1(NIII)
      DO 216JJJ=2,NIII
216   ZZZB(JJJ)=E1(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(8)=ZZZB(NIII)
      ZZZB(1)=E2(NIII)
      DO 217JJJ=2,NIII
217   ZZZB(JJJ)=E2(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(9)=ZZZB(NIII)
      ZZZB(1)=TMP1(NIII)
      DO 218JJJ=2,NIII
218   ZZZB(JJJ)=TMP1(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(10)=ZZZB(NIII)
      ZZZB(1)=TMP2(NIII)
      DO 219JJJ=2,NIII
219   ZZZB(JJJ)=TMP2(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(11)=ZZZB(NIII)
      ZZZB(1)=XK1(NIII)
      DO 220JJJ=2,NIII
220   ZZZB(JJJ)=XK1(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(12)=ZZZB(NIII)
      ZZZB(1)=XK2(NIII)
      DO 221JJJ=2,NIII
221   ZZZB(JJJ)=XK2(NIII-JJJ+1) + T*ZZZB(JJJ-1)
      YNEW(13)=ZZZB(NIII)
      RETURN
      END
```

References

1. J. E. Fromm, "The Time-dependent Flow of an Incompressible Fluid," Methods in Comput. Physics, 3, New York, p. 346 (1964).

2. R. D. Richtmeyer and K. W. Morton, Difference Methods For Initial Value Problems, Interscience, New York (1967).

3. O. C. Zienkiewicz, The Finite Element Method, Third Edition, McGraw-Hill, London (1977).

4. R. L. Brown, "Investigation of the Computational Aspects of the Numerical Solution of Flow on a Cone." Report 22, Vol. II, Research Reports of 1978 USAF-ASEE Faculty Research Program, (1978).

5. S. C. Lubard, and W. S. Helliwell,"Calculation of the Flow on a Cone at High Angle of Attack," pp. 5-12, RDA-TR-150, R&D Associates, Santa Monica, CA (1973).

6. J. Douglas, and G. E. Gunn,"A General Formulation of Alternating Direction Methods," Num. Math., 6, pp. 228-253 (1964).

7. R. M. Beam, and R. F. Warming, "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form," Journal Computational Physics, 22, pp. 87-110 (1976).

8. D. W. Peacemen, and H. H. Rachford, "The Numerical Solution of Parabolic and Elliptical Differential Equations," SIAM Journal, 3, pp. 28-41 (1955).

9. W. S. Helliwell, and S. C. Lubard, "An Implicit Method for Three-Dimensional Viscous Flow with Application to Cone at Angle of Attack," Report TR-0074(4450-64)-1, The Aerospace Corporation, Santa Monica, CA (1973).

10. J. M. Hyman, "A Method of Lines Approach to the Numerical Solution of Conservation Laws," Advances In Computer Methods for Partial Differential Equations III, pp. 313-321, R. Vichnevetsky and R. S. Stepleman (ed.) IMACS (1979).

11. G. Dahlquist, "A Special Stability Problem for Linear Multistep Methods," BIT, 3, pp. 27-43 (1963).

12. S. C. Lubard and W. S. Helliwell, "Calculation of the Flow on a Cone at High Angle of Attack," pp. 26-30, RDA-TR-150, R&D Associates Santa Monica, CA (1973).

13. R. M. Beam and R. F. Warming, "On the Construction and Application of Implicit Factored Schemes for Conservation Laws," SIAM-AMS Proceedings of Symposium on Computational Fluid Dynamics, Vol. II (1977).

14. R. L. Brown, "Stability of Sequences Generated by Nonlinear Differential Systems," Math. Comp., 33, pp. 637-645 (1979).

15. G. Dahlquist, "G-stability is Equivalent to A-stability," BIT, 18, pp. 384-401 (1979).

16. N. Rouche, P. Habets, and M. Laloy, Stability Theory by Liapunov's Direct Method, Springer-Verlag, New York (1977).

17. K. R. Kovach, "A Precompiler for Deriving the Time Series Solution to Systems of Differential Equations," M.S. Thesis, University of Virginia, Charlottesville, VA (1980).

18. D. H. Norrie, G. de Vries, The Finite Element Method, Academic Press, New York (1973).

19. A. R. Mitchell, R. Wait, The Finite Element Method in Partial Differential Equations, John Wiley & Sons, New York (1977).

20. A. C. Hindmarsh, "The LLL Family of Ordinary Differential Equation Solvers," UCRL-78129 (1976).

21. G. D. Byrne, A. C. Hindmarsh, "A Polyalgorithm for the Numerical Solution of Ordinary Differential Equations," ACM TOMS, 6 (1975).

DISTRIBUTION LIST

Copy No.	
1 - 16	Air Force Office of Scientific Research Bolling Air Force Base Washington, D. C. 20332
17 - 18	R. L. Brown
19	J. M. Ortega
20	I. A. Fischer Office of Sponsored Programs
21 - 22	E. H. Pancake Clark Hall
23	RLES Files

0692:jt

UNIVERSITY OF VIRGINIA

School of Engineering and Applied Science

The University of Virginia's School of Engineering and Applied Science has an undergraduate enrollment of approximately 1,000 students with a graduate enrollment of 350. There are approximately 120 faculty members, a majority of whom conduct research in addition to teaching.

Research is an integral part of the educational program and interests parallel academic specialties. These range from the classical engineering departments of Chemical, Civil, Electrical, and Mechanical to departments of Biomedical Engineering, Engineering Science and Systems, Materials Science, Nuclear Engineering, and Applied Mathematics and Computer Science. In addition to these departments, there are interdepartmental groups in the areas of Automatic Controls and Applied Mechanics. All departments offer the doctorate; the Biomedical and Materials Science Departments grant only graduate degrees.

The School of Engineering and Applied Science is an integral part of the University (approximately 1,400 full-time faculty with a total enrollment of about 14,000 full-time students), which also has professional schools of Architecture, Law, Medicine, Commerce, and Business Administration. In addition, the College of Arts and Sciences houses departments of Mathematics, Physics, Chemistry and others relevant to the engineering research program. This University community provides opportunities for interdisciplinary work in pursuit of the basic goals of education, research, and public service.

FILMED

8